



Universidad
Carlos III de Madrid

Departamento de Informática

PROYECTO FIN DE CARRERA

SIGA, Sistema Integral para la Gestión de la Accesibilidad

Autor: Adrián López Pérez

Tutor: Lourdes Moreno López

Leganés, Julio de 2010

Título: SIGA, Sistema Integral para la Gestión de la Accesibilidad

Autor: Adrián López Pérez

Director: Lourdes Moreno López

EL TRIBUNAL

Presidente: _____

Vocal: _____

Secretario: _____

Realizado el acto de defensa y lectura del Proyecto Fin de Carrera el día ____ de _____
de 20__ en Leganés, en la Escuela Politécnica Superior de la Universidad Carlos III de
Madrid, acuerda otorgarle la CALIFICACIÓN de

VOCAL

SECRETARIO

PRESIDENTE

Agradecimientos

A mi familia, amigos, compañeros de trabajo y fatiga, especialmente a todas las personas que a lo largo de mi carrera me han apoyado y ayudado a seguir adelante un día tras otro.

Una mención especial a mis padres, José Antonio y María Josefa, sin ellos nada de esto habría sido posible, a mi hermano Rubén por su apoyo moral, a Paco por su constante compañía, a Alba que me ha ayudado muchísimo al aguantarme con mis aburridísimas charlas y excentricidades y animando a seguir siempre adelante, a Lourdes que me introdujo en el apasionante mundo de la accesibilidad y siempre me ha orientado y ayudado de forma excepcional, a Esperanza y Enrique que me han escuchado en los malos momentos y han mejorado los buenos con su compañía y por último pero no menos importante a Daniel, quien ha sido mi incansable compañero de fatigas desde las primeras prácticas de la carrera, con quien he compartido las ganas de abandonarlo todo y la ilusión de seguir adelante y con quien he conseguido aprender a sufrir sin dejar de sonreír.

Muchas gracias a todos, sin vosotros esto no hubiera sido posible.

Resumen

A día de hoy Internet se ha convertido en una herramienta indispensable para un amplio número de personas de nuestra sociedad. A través de este medio de comunicación muchos usuarios acceden a un conjunto de servicios cada vez más amplio y extendido. Sin embargo existen grupos de usuarios a los que les resulta imposible o que tienen problemas con el acceso a estos servicios debido a la existencia de importantes barreras de accesibilidad.

Entre estos grupos de usuarios están las personas que padecen algún tipo de discapacidad que les dificulte el acceso normal a un equipo informático o a los propios contenidos, sin embargo no queda limitado a ellos, ya que las personas sin conocimientos sobre uso de Internet, personas de edad avanzada o gente que debido a cualquier causa tenga temporalmente mermadas sus capacidades, puede tener los mismos problemas de acceso que las personas con discapacidad.

Las limitaciones que estas dificultades suponen para la navegación por ciertos servicios de la red además de suponer un problema grave para las personas afectadas, también es una vulneración de sus derechos, ya que existen actualmente tanto un conjunto de convenciones internacionales, como legislaciones nacionales que dictan el derecho de todas las personas al acceso a la información y a la no discriminación.

La aplicación SIGA, Sistema Integral para la Gestión de la Accesibilidad proporciona una herramienta para que empresas o entidades que dispongan de un sitio web con una complejidad media o alta puedan realizar una monitorización completa de la accesibilidad de sus páginas, aumentando notablemente su calidad y prestando sus servicios a un número de usuarios potencialmente mayor. Este sistema está orientado para ser usado tanto por personas individuales como por grupos dedicados al desarrollo y mantenimiento, los cuales podrán obtener un beneficio mayor.

La aplicación proporcionará un subsistema de gestión de incidencias de los usuarios del sitio, orientado a ayudar a las personas que hayan encontrado problemas en la navegación y a evitar que estos problemas se repitan en el futuro; un subsistema dedicado análisis automático de problemas de accesibilidad y un subsistema de gestión del conocimiento que permitirá almacenar y clasificar instancias de conocimiento por los miembros del equipo.

Como detalle de arquitectura, la aplicación está diseñada para funcionar tanto en el mismo servidor, siempre que se adapte a los requisitos, como en un servidor diferente al del sitio o sitios web que se deseen monitorizar, proporcionando independencia de las tecnologías ya

utilizadas. Adecuándose a los subsistemas anteriormente indicados la aplicación se divide dos partes, una interfaz pública, mediante la cual un usuario anónimo puede crear incidencias en el sistema y una interfaz privada que permite el acceso a los gestores del sistema y proporciona los módulos para gestión de las incidencias, visualización de alertas automáticas, gestión del conocimiento y administración del sistema.

Abstract

Nowadays Internet has become an indispensable tool for a large number of people in our society. Through this medium many users access to a range of services increasingly broad and wide. However, there are groups of users that are unable or have problems with the access to these services due to the existence of significant barriers to accessibility.

Among these groups of users are people with some type of disability that hinders normal access to a computer or the web contents, but not only people with disabilities, because users without knowledge of Internet use, old people or people due to any cause has temporarily limited their abilities, may have the same problems of access that people with disabilities.

The constraints that these difficulties means for navigation on certain network services not only suppose a serious problem for those affected, is also a violation of their rights as they currently exist as a set of international conventions, such as national laws that they dictate the right of all the people to access the information and to the not discrimination.

The application provides a tool for companies or entities that have a website with a medium or high complexity. A full monitoring of the accessibility of their pages can be made, significantly increasing its quality and to serve a potentially larger number of users. This system is intended to be used both by individuals and by groups engaged in the development and maintenance, which may get a higher benefit.

The application will provide an incident management subsystem of site users, aimed at helping people who have encountered problems in navigation and prevent these problems from recurring in the future, a subsystem dedicated to the automatic analysis of accessibility issues and knowledge management subsystem that can store and classify items of knowledge by team members.

As an architectural detail, the application is designed to work on the same server, whenever it adapts to the requirements, as on a different server to the site o sites that want to monitor, providing independence from the technologies already used. Adapting to the subsystems listed above, the application is divided into two parts, a public interface through which an anonymous user can create incidents on the system and a private interface that allows access to system managers and provides the modules for the management of the incidents, automatic alert visualization, knowledge management and system administration.

Índice general

Capítulo 1. Introducción y objetivos.....	18
1.1. Introducción	18
1.2. Objetivos.....	19
1.3. Fases de desarrollo	19
1.4. Medios empleados	21
1.5. Estructura de la memoria.....	22
Capítulo 2. Estado de la cuestión	24
2.1. La accesibilidad web.....	24
2.2. Legislación y estándares.....	24
2.3. Dificultades en la realización y mantenimiento de un sitio accesible.	27
2.4. Estado actual de la accesibilidad web en España	27
2.5. Estandarización. Web Accessibility Initiative (WAI)	28
2.6. Técnicas de evaluación de la accesibilidad.....	30
2.7. Metodología de evaluación recomendada	31
2.8. Herramientas.....	31
2.9. Deficiencias de las herramientas actuales.....	33
2.10. Trabajos relacionados.....	33
2.11. Marco metodológico para accesibilidad	34
2.12. Tecnologías para el desarrollo de aplicaciones web	34
2.13. Entornos de desarrollo para aplicaciones web	35
2.14. Conclusiones	36
Capítulo 3. Análisis	37
3.1. Determinación del alcance del sistema.....	37
3.2. Especificación de estándares y normas.....	39
3.3. Establecimiento de requisitos de usuario.....	40
3.4. Especificación de casos de uso.....	44
3.5. Establecimiento de requisitos software	48
3.6. Matriz de trazabilidad de requisitos de software frente a requisitos de usuario	54
3.7. Diagramas de estado.....	55
Capítulo 4. Diseño	57
4.1. Tecnologías utilizadas	57
4.2. Arquitectura	58

4.3. Pautas de diseño.....	61
4.4. Diseño detallado.....	61
4.5. Diagramas de secuencia	80
Capítulo 5. Manual de usuario.....	92
5.1. Controles comunes.....	92
5.2. Acceso a la aplicación	93
5.3. Pantalla de bienvenida de la aplicación.....	94
5.4. Gestión de incidencias	94
5.5. Creación de una nueva incidencia	98
5.6. Visualización de alertas automáticas de accesibilidad	99
5.7. Gestión de la base de conocimiento	102
5.8. Administración del sistema.....	106
Capítulo 6. Validación	112
6.1. Tiempo empleado	113
6.2. Valoración en el cuestionario	114
6.3. Observaciones del cuestionario	115
6.4. Conclusiones	115
Capítulo 7. Conclusiones y Líneas futuras	116
7.1. Conclusiones	116
7.2. Líneas futuras	117
Presupuesto	122
Glosario	125
Bibliografía	127
Anexo A. Detalle de las clases JavaBean.....	131
Anexo B. Detalle de los JSP.....	138
Anexo C. Detalle de las clases de acción	145
Anexo D. Detalle de los interceptores	159
Anexo E. Detalle de clases auxiliares.....	161
Anexo F. Ejercicio de validación	163
Anexo G. Cuestionarios de validación.....	165

Índice de figuras

Figura 1-1: Ciclo de vida de un proyecto software con ESA Lite (ESA BSSC, 1996).....	20
Figura 2-1: Relación de componentes relacionados con la accesibilidad. (W3C, 2010)	29
Figura 2-2: Ciclo de implementación de herramientas. (W3C, 2010)	29
Figura 3-1: Diagrama de permisos y usuarios	38
Figura 3-2: Diagrama de casos de uso, inclusión de incidencias desde el formulario web.....	45
Figura 3-3: Diagrama de casos de uso, usuario no autenticado	45
Figura 3-4: Diagrama de casos de uso servidor, usuario con permisos de visualización de incidencias.....	46
Figura 3-5: Diagrama de casos de uso servidor, usuario con permisos de creación de incidencias.....	46
Figura 3-6: Diagrama de casos de uso servidor, usuario con permisos de creación de incidencias.....	47
Figura 3-7: Diagrama de casos de uso servidor, usuario con permisos de creación de incidencias.....	47
Figura 3-8: Diagrama de clases de uso servidor, rol administrador.....	48
Figura 3-9: Diagrama de estados, inserción de incidencia	55
Figura 3-10: Diagrama de estados, revisión de incidencias.....	56
Figura 4-1: Arquitectura del sistema	59
Figura 4-2: Arquitectura del servidor	60
Figura 4-3: Diagrama de clases, vista global	62
Figura 4-4: Módulos del sistema	63
Figura 4-5: Diagrama de clases, módulo principal y de autenticación	64
Figura 4-6: Diagrama de clases, módulo de incidencias.....	64
Figura 4-7: Diagrama de clases, módulo de alertas automáticas.....	65
Figura 4-8: Diagrama de clases, módulo de conocimiento.....	65
Figura 4-9: Diagrama de clases, módulo de administración.....	66
Figura 4-10: Diagrama de funcionamiento del <i>framework</i> Struts2	67
Figura 4-11: Diagrama de negocio para el procesamiento de una acción por Struts 2	68
Figura 4-12: Detalle elementos Action y ActionSupport	71
Figura 4-13: Diagrama de los mapeos de Struts para la aplicación	72
Figura 4-14: Diagrama relacional de la base de datos	73
Figura 4-15: Arquitectura Hibernate de alto nivel (Red Hat, Inc, 2010)	74
Figura 4-16: Arquitectura Hibernate detallada (Red Hat, Inc, 2010)	75
Figura 4-17: Diagrama de clases, interceptores	78
Figura 4-18: Diagrama de clases del analizador de accesibilidad.....	80
Figura 4-19: Obtención de sesión de Hibernate	80
Figura 4-20: Diagrama de secuencia, añadir incidencia, usuario anónimo	81
Figura 4-21: Diagrama de secuencia, añadir incidencia, usuario experto	82
Figura 4-22: Diagrama de secuencia, autenticación de usuarios	83
Figura 4-23: Diagrama de secuencia, autenticación de usuarios	84
Figura 4-24: Diagrama de secuencia, listado de incidencias.....	85
Figura 4-25: Diagrama de secuencia, visualización de incidencia.....	86
Figura 4-26: Diagrama de secuencia, edición de incidencia	87

Figura 4-27: Diagrama de secuencia, ver alertas automáticas, listado de sitios monitorizados	88
Figura 4-28: Diagrama de secuencia, ver alertas automáticas, listado de puntos de verificación	89
Figura 4-29: Diagrama de secuencia, ver alertas automáticas, listado de puntos de verificación	90
Figura 4-30: Diagrama de secuencia, ver alertas automáticas, visualizar alerta de accesibilidad	91
Figura 5-1: Controles comunes	93
Figura 5-2: Mensaje de confirmación	93
Figura 5-3: Mensaje de error	93
Figura 5-4: Interfaz de acceso a la aplicación	94
Figura 5-5: Interfaz de bienvenida	94
Figura 5-6: Interfaz de gestión de incidencias	95
Figura 5-7: Detalle de acceso a visualización y edición de incidencias	95
Figura 5-8: Interfaz de visualización y edición de incidencias	96
Figura 5-9: Detalle de estado de incidencias	97
Figura 5-10: Detalle de creador de incidencia (creador experto)	97
Figura 5-11: Detalle de creador de incidencia (acceso público).....	97
Figura 5-12: Detalle de comentarios de la incidencia	97
Figura 5-13: Interfaz de creación de incidencias (usuario experto)	98
Figura 5-14: Mensaje de incidencia creada correctamente.....	98
Figura 5-15: Proceso de visualización de alertas automáticas	99
Figura 5-16: Interfaz de selección de sitio web monitorizado.....	100
Figura 5-17: Interfaz de selección de punto de verificación	100
Figura 5-18: Interfaz de selección de alerta automática	101
Figura 5-19: Interfaz de detalle de alerta automática.....	102
Figura 5-20: Procesos de gestión del conocimiento	103
Figura 5-21: Interfaz de gestión del conocimiento	103
Figura 5-22: Detalle del árbol de categorías.....	104
Figura 5-23: Interfaz de listado de artículos de conocimiento por categorías	104
Figura 5-24: Interfaz de visualización y edición de artículos de conocimiento.....	105
Figura 5-25: Interfaz de selección de categorías asociadas	105
Figura 5-26: Interfaz de administración	106
Figura 5-27: Diagrama de estructura de permisos.....	107
Figura 5-28: Interfaz de administración de roles y permisos	108
Figura 5-29: Interfaz de modificación de roles	108
Figura 5-30: Interfaz de administración de usuarios y roles	109
Figura 5-31: Interfaz de modificación de usuarios	109
Figura 5-32: Interfaz de administración de estados de incidencias	110
Figura 5-33: Interfaz de administración de categorías.....	111
Figura 5-34: Interfaz de administración de sitios monitorizados	111
Figura 6-1: Gráfico de tiempo empleado por tareas.....	113
Figura 6-2: Gráfico de tiempo empleado por usuario en cada tarea	114
Figura 6-3: Media de valoraciones en las preguntas del cuestionario	114

Figura 6-4: Varianza de valoraciones en las preguntas del cuestionario	115
Figura 7-1: Diagrama de Gantt para las tareas Investigación del estado del arte, Realización del presupuesto y Análisis	122
Figura 7-2: Diagrama de Gantt para la tarea Diseño	122
Figura 7-3: Diagrama de Gantt para la tarea Implementación	123
Figura 7-4: Diagrama de Gantt para la tarea Validación	123
Figura 7-5: Diagrama de Gantt para las tareas Creación del manual de usuario, Definición de líneas futuras, Redacción de la memoria del proyecto, Creación de la presentación del proyecto y Validación general	123

Índice de tablas

Tabla 3-1: RU01	41
Tabla 3-2: RU02	41
Tabla 3-3: RU03	41
Tabla 3-4: RU04	41
Tabla 3-5: RU05	42
Tabla 3-6: RU06	42
Tabla 3-7: RU07	42
Tabla 3-8: RU08	42
Tabla 3-9: RU09	42
Tabla 3-10: RU10	43
Tabla 3-11: RU11	43
Tabla 3-12: RU12	43
Tabla 3-13: RU13	43
Tabla 3-14: RU14	43
Tabla 3-15: RU15	44
Tabla 3-16: RU16	44
Tabla 3-17: RFunc01	48
Tabla 3-18: RFunc02	49
Tabla 3-19: RFunc03	49
Tabla 3-20: RFunc04	49
Tabla 3-21: RFunc05	49
Tabla 3-22: RFunc06	49
Tabla 3-23: RFunc07	50

Tabla 3-24: RFunco8	50
Tabla 3-25: RFunco9	50
Tabla 3-26: Rfunc10	50
Tabla 3-27: Rfunc11	50
Tabla 3-28: Rfunc12	51
Tabla 3-29: Rfunc13	51
Tabla 3-30: Rfunc14	51
Tabla 3-31: RDesemp01	51
Tabla 3-32: RInterfo1	52
Tabla 3-33: RInterfo2	52
Tabla 3-34: RInterfo3	52
Tabla 3-35: ROpero1	52
Tabla 3-36: RVerifo1	52
Tabla 3-37: RVerifo2.....	53
Tabla 3-38: RVerifo2	53
Tabla 3-39: RDocu01.....	53
Tabla 3-40: RSeguo1.....	53
Tabla 3-41: RSeguo2	53
Tabla 3-42: RCalo1.....	54
Tabla 3-43: REntro1	54
Tabla 3-44: Matriz de trazabilidad requisitos software – requisitos de usuario.....	54
Tabla 7-1: LFo1	118
Tabla 7-2: LFo2	118
Tabla 7-3: LFo3	118
Tabla 7-4: LFo4	118
Tabla 7-5: LFo5	119
Tabla 7-6: LFo6	119
Tabla 7-7: LFo7.....	119
Tabla 7-8: LFo8	119
Tabla 7-9: LFo9.....	120
Tabla 7-10: LF10	120
Tabla 7-11: LF11	120
Tabla 7-12: LF12.....	120
Tabla 7-13: LF13.....	120
Tabla 7-14: LF14	121
Tabla A-1: Detalle del JavaBean AutomaticWarningBean.....	131
Tabla A-2: Detalle del JavaBean ClassificationBean.....	132
Tabla A-3: Detalle del JavaBean CommentBean	132
Tabla A-4: Detalle del JavaBean IncidenceBean	133
Tabla A-5: Detalle del JavaBean IncidenceStateBean	134
Tabla A-6: Detalle del JavaBean KnowledgeBean.....	134
Tabla A-7: Detalle del JavaBean KnowledgeClassificationBean	135
Tabla A-8: Detalle del JavaBean RoleBean	135
Tabla A-9: Detalle del JavaBean SiteBean	136
Tabla A-10: Detalle del JavaBean UserBean	136

Tabla B-1: Leyenda de detalle de JSP	138
Tabla B-2: Detalle del JSP admin.jsp	138
Tabla B-3: Detalle del JSP adminClassification.jsp	138
Tabla B-4: Detalle del JSP adminRoles.jsp	139
Tabla B-5: Detalle del JSP adminSites.jsp	139
Tabla B-6: Detalle del JSP adminStates.jsp	139
Tabla B-7: Detalle del JSP adminUsers.jsp	139
Tabla B-8: Detalle del JSP createExpertIncidence.jsp	140
Tabla B-9: Detalle del JSP createIncidence.jsp	140
Tabla B-10: Detalle del JSP createKnowledge.jsp	140
Tabla B-11: Detalle del JSP login.jsp	140
Tabla B-12: Detalle del JSP menu.jsp	141
Tabla B-13: Detalle del JSP myIncidences.jsp	141
Tabla B-14: Detalle del JSP showCheckpoint.jsp	141
Tabla B-15: Detalle del JSP showClassifKnowledge.jsp	141
Tabla B-16: Detalle del JSP showIncidence.jsp	142
Tabla B-17: Detalle del JSP showKnowledge.jsp	142
Tabla B-18: Detalle del JSP showSites.jsp	142
Tabla B-19: Detalle del JSP showSpecificKnowledge.jsp	142
Tabla B-20: Detalle del JSP showWarn.jsp	143
Tabla B-21: Detalle del JSP showWarns.jsp	143
Tabla B-22: Detalle del JSP start.jsp	143
Tabla B-23: Detalle del JSP successCommit.jsp	143
Tabla B-24: Detalle del JSP successCommitExpert.jsp	144
Tabla B-25: Detalle del JSP updateRoles.jsp	144
Tabla B-26: Detalle del JSP updateStates.jsp	144
Tabla B-27: Detalle del JSP updateUsers.jsp	144
Tabla C-1: Leyenda de detalle de la clase de acción.	145
Tabla C-2: Detalle de la clase de acción AdminClassification	145
Tabla C-3: Detalle de la clase de acción AdminRoles.....	146
Tabla C-4: Detalle de la clase de acción AdminSites	148
Tabla C-5: Detalle de la clase de acción AdminStates	149
Tabla C-6: Detalle de la clase de acción AdminUsers	150
Tabla C-7: Detalle de la clase de acción IncidenceAction	151
Tabla C-8: Detalle de la clase de acción IncidenceBean	152
Tabla C-9: Detalle de la clase de acción KnowledgeAction	153
Tabla C-10: Detalle de la clase de acción ListIncidences.....	155
Tabla C-11: Detalle de la clase de acción Login	155
Tabla C-12: Detalle de la clase de acción Logout	156
Tabla C-13: Detalle de la clase de acción MenuInterface	156
Tabla C-14: Detalle de la clase de acción StartInterface.....	156
Tabla C-15: Detalle de la clase de acción WarningAction	157
Tabla D-1: Interceptor dbInterceptor.....	159
Tabla D-2: Interceptor loginInterceptor.....	159
Tabla D-3: Interceptor adminInterceptor	159

Tabla D-4: Interceptor autoWarningInterceptor	159
Tabla D-5: Interceptor createIncidentesInterceptor	160
Tabla D-5: Interceptor dbInterceptor	160
Tabla D-6: Interceptor knowledgeInterceptor	160
Tabla D-7: Interceptor showIncidentesInterceptor	160
Tabla E-1: Detalle del enum EnumMessages	161
Tabla E-2: Detalle de la clase InitSessionFactory	161
Tabla E-3: Detalle de la clase SessionContainer	162
Tabla E-4: Detalle de la clase Util	162

Capítulo 1.

Introducción y objetivos

1.1. Introducción

Este proyecto recoge todo el proceso de desarrollo de la aplicación SIGA, Sistema Integral para la Gestión de la Accesibilidad, este software tiene como objetivo proporcionar una herramienta de apoyo a desarrolladores de sitios web y a las entidades interesadas en proporcionar unos servicios web de calidad, facilitando el mantener la accesibilidad de los sitios web a un alto nivel.

El panorama web actual, con su creciente evolución, la aparición de la web 2.0 y la integración del visitante del sitio como una pieza más para la creación de éste ha provocado la aparición de nuevas necesidades y horizontes alrededor de los portales tanto personales como empresariales u organizacionales. La creación de contenido en estos portales ha pasado de ser una tarea destinada a una persona a ser una colaboración entre el propietario del sitio y sus usuarios.

Actualmente las especificaciones sobre accesibilidad así como la legislación vigente que afecta a gran cantidad de sitios web, indican que todo sitio web además de cumplir con los criterios de accesibilidad establecidos debe disponer de un sistema que permita a cualquier usuario notificar fallos en la accesibilidad, usabilidad o cualquier otro aspecto del sitio, y que estas notificaciones sean atendidas y procesadas. Para cumplir este requisito se puede proporcionar simplemente una dirección de email, donde se puedan reportar los problemas de accesibilidad detectados, sin embargo se recomienda hacer uso de un sistema más automatizado que ayude a su gestión (Moreno López, 2010), tal como se recoge en el marco metodológico AWA recogido como el mecanismo de calidad de la accesibilidad CAL 01.03/01.

Este sistema y los procedimientos asociados a él adquieren mayor importancia en sitios web 2.0, donde los usuarios pueden realizar sus propias aportaciones y teniendo en cuenta que no siempre pueden ser revisadas antes de su publicación y se convierten en una fuente potencial de problemas de accesibilidad; obsérvese como ejemplo una página donde se permite a los usuarios incluir sus propios artículos, al hacerlo podrían no especificar los textos alternativos en las imágenes y, aunque esto se podría controlar automáticamente, se podrían insertar textos que no satisficieran las necesidades de este campo no aportando la información

necesaria, lo que es prácticamente imposible de controlar con un mecanismo automático. Por lo tanto la calidad de estos sitios web puede verse gradualmente reducida según se publique nuevo contenido que no haya sido correctamente revisado, sin la posibilidad de formar a estos publicadores ya que son usuarios del sitio y no personas dedicadas a su creación.

1.2. Objetivos

Se pretende diseñar un sistema que permita la monitorización de la accesibilidad de múltiples sitios web, presentando un conjunto de servicios que faciliten la tarea de mantener un alto nivel de accesibilidad. Entre estos servicios se destaca la posibilidad de que los usuarios puedan notificar fallos o problemas de accesibilidad, la obtención automatizada y presentación de alertas a partir de un analizador de accesibilidad automático y un sistema de gestión del conocimiento adquirido. El conjunto del sistema pretende ser una herramienta integral de gestión que permita aumentar la calidad del sitio web, reduciendo en la medida de lo posible el esfuerzo humano necesario para la tramitación y solución de las incidencias y las alertas detectadas.

Los principales objetivos se listan a continuación:

- Como funcionalidades base a desarrollar, se establecen un sistema de comunicación de incidencias por parte de los usuarios y de gestión de estas en el sistema, un subsistema dedicado análisis automático de problemas de accesibilidad y un subsistema de gestión del conocimiento que permitirá almacenar y clasificar instancias de conocimiento por los miembros del equipo.
- El sistema debe permitir una fácil y rápida implantación en cualquier sitio web, presentándose al usuario mediante un enlace, desde al que se accede a la página de creación de incidencias. El portal dedicado a la administración dispondrá de un acceso web desde el que se proporcionará toda la funcionalidad necesaria para el completo manejo.
- Para conseguir una mayor independencia de plataforma el sistema se implementará en módulos completamente diferenciados, los cuales comprenderán, el interfaz de creación de incidencias, el cual deberá poder incluirse fácilmente en cualquier sitio con la máxima independencia de tecnología; el motor de persistencia, el cual deberá poder adaptarse a los distintos entornos sin afectar al resto de la aplicación y por lo tanto se deberá poder modificar el software para el funcionamiento con cualquier gestor de base de datos con un coste mínimo; y el interfaz de gestión y administración y el motor del sistema.
- La interfaz de usuario tanto pública como privada deberá presentar un alto nivel de usabilidad y accesibilidad, cumpliendo al menos con el nivel de prioridad doble-A de WCAG 2.0 y adoptando un enfoque de diseño centrado en el usuario (DCU). De esta forma el interfaz resultante presentará la alta calidad necesaria.

1.3. Fases de desarrollo

Con el objetivo de seguir un proceso estandarizado se han aplicado dos metodologías al desarrollo de este proyecto, estas se detallan a continuación:

1.3.1. Metodología ESA lite

Para el desarrollo de la aplicación se ha utilizado la metodología ESA Lite (ESA BSSC, 1996). Esta metodología ha sido desarrollada por la Agencia Espacial Europea como estándar para el desarrollo de proyectos software y está orientada a proyectos de menor envergadura que el estándar original de ESA, siendo los límites relativos definidos (ESA BSSC, 1996):

- Menos de dos años / persona de desarrollo.
- Un único equipo de desarrollo de 5 ó menos personas.
- La cantidad de código fuente producida debe ser inferior a 10.000 líneas de código, excluyendo comentarios.

Las ventajas de adoptar esta metodología orientada a proyectos pequeños en vez de una más pesada como ESA o Métrica 3 son (ESA BSSC, 1996):

- Combinar las fases de requisitos software y diseño de la arquitectura.
- Simplificar la documentación.
- Simplificar la planificación.
- Reducir los requisitos de confiabilidad.
- Utilizar la especificación de prueba del sistema para las pruebas de aceptación.

El ciclo de vida del desarrollo de una aplicación con ESA Lite se corresponde con el presentado en la Figura 1-1.

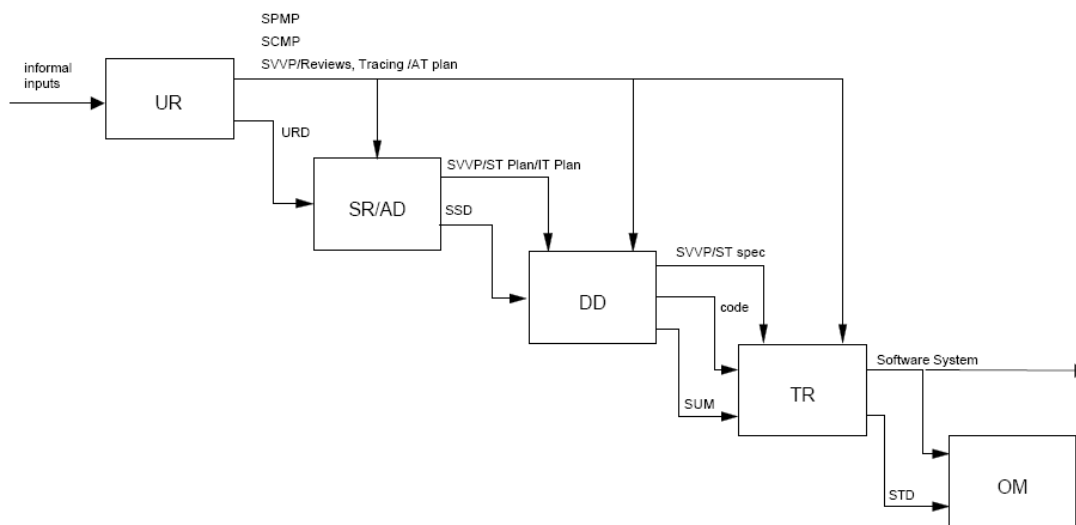


Figura 1-1: Ciclo de vida de un proyecto software con ESA Lite (ESA BSSC, 1996)

Identificación de Las fases del desarrollo en este documento

Debido a que la estructura del presente documento sigue un patrón diferente al definido por ESA Lite, el cual se compone de varios documentos separados como se puede observar en las salidas de los procesos en la Figura 1-1, se indica la correspondencia entre los documentos de ESA Lite y los apartados del presente documento:

- UR (Requisitos de usuario): Corresponde a los apartados 3.1 Determinación del alcance del sistema, 3.2 Especificación de estándares y normas y 3.3 Establecimiento de requisitos de usuario..
- SR/AD (Requisitos software / Diseño de la arquitectura): Corresponde a los apartados 3.4 Especificación de casos de uso, 3.5 Establecimiento de requisitos software, 3.6 Matriz de trazabilidad de requisitos de software frente a requisitos de usuario, 3.7 Diagramas de estado, 4.1 Tecnologías utilizadas y 4.2 Arquitectura.
- DD (Diseño detallado): Corresponde a los apartados 4.3 Pautas de diseño, 4.4 Diseño detallado y 4.5 Diagramas de secuencia.
- TR (Pruebas de validación y verificación): Corresponde al apartado Capítulo 6 Validación.
- OM (Mantenimiento del sistema): Al no implantarse en el momento presente el sistema no se ha realizado un programa de mantenimiento. Este se trataría en caso de producirse esta implantación.

1.3.2. Accessibility for Web Applications

Debido a la necesidad de que la aplicación cumpla con los estándares de Web Content Accessibility Guidelines (WCAG) 2.0 y la complejidad que esta funcionalidad puede conllevar, se ha decidido aplicar el marco metodológico AWA (Accessibility for Web Applications) (Moreno López, 2010), de esta manera se facilita la implantación de la accesibilidad desde el diseño de la aplicación, evitando el elevado coste que conllevaría incluir la accesibilidad en las últimas etapas del desarrollo. Para ello se han identificado los requisitos AWA con los que esta aplicación tendrá relación y se han aplicado sus mecanismos derivados al diseño del sistema. De esta forma se puede garantizar que la accesibilidad final de las páginas web pertenecientes al sistema tendrá un muy alto nivel, cumpliendo con los requisitos mínimos en relación a la accesibilidad para esta aplicación.

Debido a la naturaleza de la aplicación, solo ha sido necesario aplicar los mecanismos de accesibilidad AWA en el módulo de la vista, ya que tanto el controlador como el modelo son totalmente independientes.

1.4. Medios empleados

Para la realización del proyecto se ha hecho uso de los siguientes recursos hardware y software:

1.4.1. Recursos hardware

Para la realización del software y la memoria se han utilizado 2 equipos informáticos con las siguientes características:

- Portátil Apple MacBook 5,1, Intel Core 2 Duo 2.4 GHz, 4GB de RAM.
- PC AMD 2600+, 1,5 GB de RAM.

1.4.2. Recursos software

El software detallado a continuación ha sido utilizado en al menos uno de los equipos anteriores:

- Mac OS X 10.5: Sistema operativo del equipo MacBook.

- Microsoft Windows XP Profesional: Sistema operativo del equipo PC.
- Eclipse IDE Galileo: Entorno de programación utilizando en ambos equipos. Se hace uso del JDK Java 1.6.
- Servidor de aplicaciones web Apache Tomcat 6, haciendo uso de las siguientes librerías adicionales así como las librerías requeridas por estas:
 - Struts 2
 - Hibernate
 - Mysql Connector
- Servidor web Apache con librerías para servir webs programadas en PHP.
- Servidor de base de datos MySQL.
- Microsoft Office 2007: Suite de ofimática para el equipo PC.
- Microsoft Office 2008: Suite de ofimática para el equipo MacBook.
- OmniGraffle Professional: Software de creación de gráficos y diagramas.
- SVN Server: Servidor de control de versiones que presenta varias ventajas frente a CVS.
- Adobe Photoshop CS4 y Gimp: Software de diseño gráfico para la creación de los iconos y las imágenes de la aplicación.
- Dropbox: Software de sincronización de ficheros similar a SVN pero no específico para código fuente. Utilizado para la compartición del fichero de la memoria entre ambos equipos.
- Navegadores web para las pruebas de la aplicación:
 - Safari
 - Mozilla Firefox
 - Google Chrome
 - Internet Explorer

1.5. Estructura de la memoria

A continuación se expone la estructura seguida en el presente documento, indicando el contenido de cada uno de los capítulos siguientes:

- Capítulo 2 Estado de la cuestión: Se analiza el panorama actual en torno a la accesibilidad web y se exploran las aplicaciones disponibles relacionadas con la actual, indicando las diferencias entre ellas. Por último se indican los trabajos relacionados con este.
- Capítulo 3 Análisis: Se especifica el análisis de la aplicación, exponiendo el alcance del sistema a desarrollar y los estándares utilizados para el resto de la documentación, se detallan los casos de uso del sistema y los requisitos de usuario que este debe cubrir y, por último se presentan los requisitos software del sistema obtenidos en base a los requisitos de usuario y la matriz de trazabilidad entre estos así como el detalle de mecanismos importantes del funcionamiento del sistema mediante diagramas de estado.
- Capítulo 4 Diseño: Este capítulo recoge desde el diseño arquitectónico hasta el de bajo nivel para la aplicación, indicando las tecnologías utilizadas para su desarrollo, las pautas del diseño seguidas y por último el detalle de implementación del sistema.

- Capítulo 5 Manual de usuario: Se presenta el manual para el manejo de la aplicación de una forma simple y concisa, ayudando a la exposición con ilustraciones de la aplicación. Este documento será genérico para todos los usuarios.
- Capítulo 6 Validación: Se exponen las pruebas realizadas al sistema para la comprobación del cumplimiento de todos los requisitos establecidos en la fase de análisis. Así mismo adoptando un enfoque de diseño centrado en el usuario (DCU), se detallan los resultados de la validación con usuarios orientada a validar el interfaz y la interacción con la aplicación.
- Capítulo 7 Conclusiones y Líneas futuras: Se exponen las conclusiones del autor a la realización de este proyecto y se presentan un conjunto de posibles líneas futuras, las cuales permitirían ampliar y mejorar la funcionalidad del sistema.
- Presupuesto: Se presenta la planificación del proyecto y el presupuesto del desarrollo, el cual incluye todos los trabajos realizados relativos a este, así como todos los gastos directos o indirectos que ha ocasionado.

Capítulo 2.

Estado de la cuestión

2.1. La accesibilidad web

En la sociedad actual, las Tecnologías de la Información y Comunicación (TIC) y en particular Internet adquieren una gran importancia, siendo un elemento clave en la vida personal y profesional de una gran parte de la población. Entre todos los servicios telemáticos disponibles a través de Internet se deben destacar los servicios web, mediante los cuales están disponibles para los usuarios un gran número de aplicaciones utilizadas en el día a día. Tareas tan cotidianas como la búsqueda de información a través de Internet, el envío y recepción de correos electrónicos mediante clientes web, herramientas administrativas, soluciones empresariales, prestación de servicios telemáticos por parte de la administración o de empresas privadas, servicios multimedia, de ocio, entretenimiento y un larguísimo etcétera son actualmente necesarias en la vida de muchos de los usuarios de la web. Sin embargo este contenido en muchos casos no está preparado para ser utilizado por todas las personas que deseen hacerlo, ya que existe una parte muy importante de la población que presenta algún tipo de discapacidad y por ello pueden encontrar dificultades al acceder a estos servicios o incluso que les resulte imposible; sin embargo no solo los usuarios con discapacidad pueden encontrar estos problemas, sino también personas que no están acostumbradas a las nuevas tecnologías o incluso que no disponen de los medios tecnológicos específicos o de acceso a internet para los que ha sido concebida la aplicación.

Desde un punto de vista social es muy importante que los grupos de usuarios con discapacidad en ningún caso sean discriminados en el uso de las aplicaciones web, y de igual forma desde un punto de vista económico para la empresa que preste el servicio, ya que la población de personas con algún tipo de discapacidad representa en nuestro país un 8,5% del total de nuestra población. Este porcentaje se cuantifica en aproximadamente 3,8 millones de personas con discapacidad en nuestro país, 50 millones a nivel europeo y alrededor de 500 millones a nivel mundial, cifras que no deberían ser despreciables para ninguna empresa o entidad que preste servicios o quiera promocionarse a través de Internet (Eurostat, 1996).

2.2. Legislación y estándares

Actualmente el uso de las tecnologías de la información es un derecho de todos, muchos países han reconocido este derecho y han creado una legislación de cara a asegurar este

acceso equitativo para todas las personas. Por lo tanto, en el momento de crear un nuevo sitio web es importante conocer y actuar cumpliendo con la legislación de cada país.

De acuerdo con estos datos y debido a la importancia en primer lugar de la “Carta Internacional de Derechos Humanos” (Naciones Unidas, 1948), la cual sentó las bases morales de igualdad se han creado legislaciones a nivel mundial en torno a las personas con discapacidad, las cuales afectan asimismo a la accesibilidad web. Cabe destacar a nivel internacional las “Normas Uniformes sobre la igualdad de oportunidades para las personas con discapacidad” (Naciones Unidas, 1993), destinada a garantizar que todos los miembros de cualquiera que sea su sociedad puedan tener los mismos derechos y obligaciones que los demás, así como la “Convención de Derechos de las Personas con Discapacidad” (Naciones Unidas, 2006), creada en Mayo de 2008, en la que “se adopta una amplia clasificación de las personas con discapacidad y se reafirma que todas las personas con todos los tipos de discapacidad deben poder gozar de todos los derechos humanos y libertades fundamentales”. Asimismo a nivel Europeo se han tomado también una serie de acciones de cara a asegurar el acceso web a las personas con discapacidad, iniciadas con el “Plan de Acción eEurope 2002” (UE, 2001), que promueve “adoptar las orientaciones de la iniciativa «Accesibilidad de la web» (WAI) para los sitios web públicos.”, y continuadas con otras acciones como eEurope 2005 (UE, 2002), donde se indica que “Debe prestarse especial atención al establecimiento de normas europeas en materia de accesibilidad a la red, de líneas directrices sobre la iniciativa de accesibilidad de la web (WAI) y de normas para un etiquetado común de las páginas web accesibles” e i2010 (UE, 2005) en la que la comisión europea propuso “publicar unas orientaciones políticas sobre accesibilidad electrónica” y orientada a la “integración de las personas con discapacidad y/o de la tercera edad en las Tecnologías de la Información y la Comunicación (TIC)”.

Estos planes y convenciones internacionales han dado paso a legislaciones nacionales que ratifican la importancia para los gobiernos de proporcionar los mismos derechos para todos los ciudadanos. Entre estas acciones cabe destacar las siguientes legislaciones que regulan la accesibilidad de las aplicaciones web:

- Sección 508 del Acta de Rehabilitación(United States Laws, 1998), en EEUU, la cual presenta unos requisitos similares a los de WCAG 1.0 (WAI, 1999).
- The Disability Discrimination Act (UK Statute Laws Database, 1995) en Reino Unido, que se basa igualmente en las pautas definidas por WAI para WCAG 1.0 (WAI, 1999).
- Barrierefreie Informationstechnik Verordnung – BITV (Decreto sobre Tecnología de la Información Libre de Barreras) (Legislación Alemana, 2002) en Alemania, la cual establece dos niveles de prioridad, P1 y P2, el primer nivel, correspondiente a un nivel Doble A de WCAG 1.0 (WAI, 1999), es obligatorio para todas las páginas gubernamentales, mientras que el segundo nivel, correspondiente al nivel Triple A de WCAG 1.0 es obligatorio para las páginas iniciales de los sitios gubernamentales (Fundación Sidar, 2007).

2.2.1. Legislación sobre accesibilidad en España

En España la legislación en relación a la accesibilidad web se encuentra recogida en las siguientes leyes y reales decretos:

- La Ley 34/2002, de 11 de julio, de Servicios de la Sociedad de la Información y de Comercio Electrónico (LSSICE) indica que se promoverá la adopción de normas de accesibilidad por los prestadores de servicios y los fabricantes de equipos y software, para facilitar el acceso de las personas con discapacidad o de edad avanzada a los contenidos digitales (BOE, 2002).
- La Ley 51/2003, de 2 de diciembre, de Igualdad de Oportunidades, no Discriminación y Accesibilidad Universal de para el desarrollo de la accesibilidad (LIONDAU) (BOE, 2003).
- El Real Decreto 1494/2007 de 12 de noviembre de 2007, por el que se aprueba el “Reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a las tecnologías, productos y servicios relacionados con la Sociedad de la Información y medios de comunicación social” (BOE, 2007), estableciendo que los sitios web de las administraciones públicas, entidades bancarias, aseguradoras, agencias de viajes, empresas de suministro de energía, empresas de transporte, etc. deben ser accesibles según la Norma UNE 139803:2004 (AENOR, 2004) en su prioridad 2, lo que se corresponde aproximadamente a la prioridad 2 de WCAG 1.0, con la salvedad de los siguientes requisitos, los cuales tienen una prioridad mayor en la Norma UNE que en WCAG 1.0 (Revilla, 2008):
 - El requisito UNE 4.4.4 con prioridad 1, correspondiente al punto de control 4.3 de WCAG 1.0, con prioridad 3 (“Identifique el idioma principal del documento”).
 - El requisito UNE 4.5.1 con prioridad 1, correspondiente al punto de control 13.1 de prioridad 2 de WCAG 1.0 (“Identifique claramente el objetivo de cada vínculo”).
 - El requisito UNE 4.3.9 con prioridad 2, correspondiente al punto de control 5.5 de prioridad 3 de WCAG 1.0 (“Proporcione resúmenes de las tablas”).
 - El requisito UNE 4.5.9 con prioridad 2, correspondiente al punto de control 9.4 de prioridad 3 de WCAG 1.0 (“Cree un orden lógico para navegar con el tabulador a través de vínculos, controles de formularios y objetos”).
- “Ratificación de la Convención de Derechos de las Personas con Discapacidad”, del 21 de abril de 2008.

Como se puede apreciar, esto hace más estricta la norma de AENOR que las pautas del WAI en las que están basadas, sobre todo de cara a la obligación de los sitios web mencionados a cumplir con la prioridad 2 de la norma UNE, lo que obligará a no solo a cumplir las pautas de WCAG 1.0 con un nivel de prioridad 2, sino también los puntos de control 4.3, 5.5 y 9.4, los cuales son de prioridad 3 en WCAG 1.0.

En base a los datos obtenidos sobre legislación en el ámbito de la accesibilidad, se puede afirmar que la normativa española al respecto es muy completa en comparación con las aplicadas en otros países que optaron por definir unos requisitos menores en la realización de sus legislaciones.

2.3. Dificultades en la realización y mantenimiento de un sitio accesible.

La creación de sitios web que cumplan con los requisitos de accesibilidad necesarios requiere un conjunto de acciones en todos los ámbitos del desarrollo web, incluyendo, desde la necesidad de contar con un experto en accesibilidad que estudie las posibles soluciones hasta adaptaciones en el conjunto organizativo de la organización y en los procesos de gestión en proyectos de gran envergadura. Por lo tanto, el desarrollo de un sitio web accesible no suele ser simple en la mayoría de los casos.

Además de estos problemas iniciales es importante destacar que, debido a las continuas modificaciones de contenido que sufren la mayoría de los sitios web, el nivel de accesibilidad alcanzado puede reducirse drásticamente con el tiempo si este nuevo contenido no cumple a su vez con los requisitos mínimos establecidos en la política de accesibilidad seguida en el desarrollo. Esto cobra especial importancia hoy en día, con la gran cantidad de sitios web llamados 2.0, en los que en muchos casos, los creadores de contenido son los propios usuarios los cuales forman comunidades de colaboración y que no cuentan necesariamente con los conocimientos en materia de diseño universal, accesibilidad o programación web. Esto hace necesario que se realice un seguimiento de los nuevos contenidos añadidos en el sitio web, estudiando su adecuación. De hecho, se puede comprobar, que si se analizan las páginas web actuales que buscan cumplir los criterios de accesibilidad, los principales problemas aparecidos en estas están relacionados con contenidos añadidos posteriormente a la realización del sitio web y, por lo tanto, previsiblemente ese tipo de errores no estarían presentes en un inicio.

2.4. Estado actual de la accesibilidad web en España

Existen numerosos estudios sobre el estado actual de la accesibilidad en todos los ámbitos, cabe destacar los resultados obtenidos en los estudios que analizan la accesibilidad web en páginas españolas y en especial los realizados por Discapnet, los cuales están recogidos en su página web (Fundación ONCE, Discapnet, 2008). Entre los datos más relevantes se encuentran los siguientes:

- Estudio (Fundación ONCE, Discapnet, 2008), que analizó los portales de los ministerios del gobierno de España, obteniendo según su métrica de análisis técnico, un resultado promedio de 55,82% de éxito entre todos ellos, siendo el mejor resultado, el obtenido por el Ministerio de Fomento, con un 86,49% y el peor el del Ministerio de Asuntos Exteriores y Cooperación, con un 0%.
- Estudio (Fundación ONCE, Discapnet, 2008), que analizó un conjunto de portales web de servicios y plataformas de eLearning, obteniendo según su métrica de análisis técnico, un resultado promedio de 20,76% de éxito, siendo el mejor resultado, el obtenido por Izanet Global Services, con un 53,85% y el único que superaba el 50% y el peor el de Smedia con un 0%.
- Estudio (Fundación ONCE, Discapnet, 2008), que analizó 18 de los portales web de los ayuntamientos de las capitales de provincia de España, obteniendo según su métrica de análisis técnico, un resultado promedio de 36,95% , siendo el mejor resultado, el obtenido por el sitio web de ayuntamiento de Madrid con un 86,49% y el peor el del ayuntamiento de Soria con un 0%

- Estudio (Rodríguez, Viera, & Castillo, 2007), que analizó una muestra de webs de universidades españolas, asegura que según los resultados del Test TAW, “el 8,3% es AAA, el 2,8% es AA y el 30,5% es A. El resto, 58,3% no alcanzan el nivel mínimo de accesibilidad”.
- Estudio (INTECO, 2008), que analizó una muestra de 1500 webs de PYMEs españolas y como conclusiones obtiene que “los sitios Web de las pymes españolas presentan importantes barreras que dificultan y/o impiden el acceso a sus contenidos por parte de cualquier persona, independientemente de las condiciones en las que se produzca dicho acceso”. Más específicamente, en un baremo establecido de 0 a 1, el índice obtenido en el estudio es de 0,49, muy inferior al umbral mínimo establecido en este estudio de 0,7, y presentando especiales deficiencias en la correcta integración de elementos de imagen, encabezados, seguimiento de las gramáticas formales, correcta configuración de los enlaces, tamaño del texto adecuado y definición de una estructura correcta.

En base a estos estudios se puede concluir que la accesibilidad web en España aunque ha sufrido una notable evolución, necesita recorrer aun un largo camino para situarse en el nivel esperado, asimismo se puede comprobar que a día de hoy, si bien la legislación vigente impulsa la adecuación de las webs a los estándares de accesibilidad, no se produce una sensación de obligación real en las entidades, existiendo muchos casos de webs obligadas por ley a ser accesibles que distan mucho de serlo.

2.5. Estandarización. Web Accessibility Initiative (WAI)

El World Wide Web Consortium (W3C) creó en 1997 la Iniciativa de Accesibilidad a la Web (WAI) (W3C, 2010), que trabaja con organizaciones de todo el mundo desarrollando estrategias, pautas y recursos de ayuda a la creación de webs accesibles para personas con discapacidad.

Según WAI, la evolución de la accesibilidad se debe realizar en todos los componentes relacionados con la Web y estos son identificados como:

- Contenido, que incluirá texto, imágenes, contenidos multimedia y el código que define la estructura de la página y la presentación.
- Navegadores de internet y reproductores, así como otros agentes de usuario.
- Tecnologías de apoyo usadas por las personas con discapacidad, como lectores de pantalla.
- El conocimiento de los usuarios.
- Desarrolladores, incluyendo a los usuarios que editan contenido.
- Herramientas de autoría.
- Herramientas de evaluación, como evaluadores de accesibilidad, validadores HTML y CSS, etc.

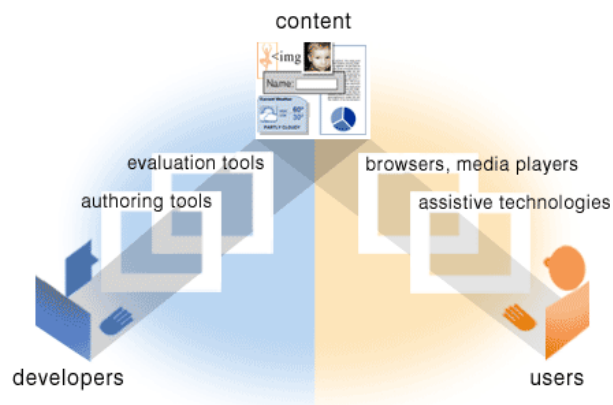


Figura 2-1: Relación de componentes relacionados con la accesibilidad. (W3C, 2010)

Todos estos componentes tienen entre ellos un alto nivel de independencia, sin embargo todos ellos afectan directamente a la accesibilidad de la web, ya que deficiencias en cualquiera de ellos podrá dificultar o impedir el acceso a grupos de usuarios.

Es importante destacar el ciclo de implementación que las herramientas para la creación y el acceso a la web experimentan en este entorno de continuo desarrollo.

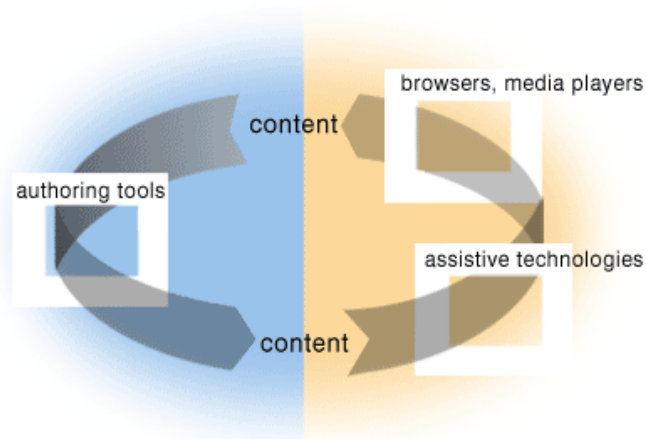


Figura 2-2: Ciclo de implementación de herramientas. (W3C, 2010)

Como se puede apreciar en la Figura 2-2, cuando se implementa el soporte a una tecnología de accesibilidad en alguna de las herramientas relacionada, las demás tecnologías tenderán a implementarla también, ya que los usuarios y desarrolladores comenzarán a demandar este soporte. Si la tecnología mejora notablemente la funcionalidad actual, previsiblemente un gran número de personas hará uso de ella y por lo tanto su implantación en las herramientas será aún mayor. WAI tiene en cuenta este proceso a la hora de la creación de nuevas pautas y estándares, ya que es básica la aceptación por parte de los usuarios y desarrolladores para obtener la máxima implantación, realizando una mayor aportación a la Web.

En la actualidad WAI ha desarrollado un conjunto de pautas y técnicas, las cuales se exponen a continuación:

- Web Content Accessibility Guidelines (WCAG) (W3C, 2008), que actualmente está en su versión 2.0: Estas pautas como hacer contenido web accesible, incluyendo texto, imágenes y contenido multimedia, así como el código web.
- How to Meet WCAG 2.0 (W3C, 2008): Una referencia rápida para los criterios de éxito y las técnicas de WCAG 2.0.
- Accessible Rich Internet Applications (WAI-ARIA) (W3C, 2009): Define la forma en la que se debe desarrollar contenido dinámico e interfaces avanzados de usuario desarrollados mediante AJAX, Javascript y otras tecnologías de forma que sean accesibles para los usuarios.
- Authoring Tool Accessibility Guidelines (ATAG) (W3C, 2010): Actualmente con borrador en su versión 2.0. Define las pautas que deben seguir las herramientas de desarrollo de sitios o páginas web.
- User Agent Accessibility Guidelines (UAAG) (W3C, 2005): Define las pautas que deben seguir los navegadores web, los reproductores y las tecnologías de apoyo.
- Evaluation and Report Language (EARL) (W3C, 2009): Define un formato independiente de plataforma para los test y las evaluaciones de accesibilidad.

2.6. Técnicas de evaluación de la accesibilidad

Para la evaluación de un sitio web existen un conjunto de técnicas que permiten conocer las características de accesibilidad. Cada una de ellas presenta diferentes ventajas y desventajas. A continuación se detallan estas técnicas:

2.6.1. Evaluación manual

Un profesional experto en accesibilidad realiza un estudio, determinando el cumplimiento de las pautas de accesibilidad por parte del sitio web.

2.6.2. Evaluación con herramientas automáticas

Consiste en la evaluación de los sitios web mediante herramientas que analizan las páginas automáticamente, indicando los fallos de accesibilidad encontrados y, los puntos que necesitan una revisión manual para determinar si cumplen o no con las pautas, y completar así la evaluación automática.

2.6.3. Evaluación basada en métricas

Para poder medir la accesibilidad de un sitio web de una forma precisa y objetiva se pueden utilizar métricas específicas. Estas métricas se definen en función de distintos parámetros y deberán proporcionar un valor de accesibilidad que normalmente será de tipo numérico y se obtendrá en base a los problemas de accesibilidad encontrados en el sitio web. Este valor permitirá comparar tanto sitios web distintos como la evolución en el tiempo de un mismo sitio web, permitiendo realizar seguimientos sencillos y bastante precisos.

2.6.4. Evaluación con usuarios

Se basa en la realización de pruebas con usuarios reales, con estas pruebas se pueden identificar problemas que no se han descubierto mediante otras técnicas como evaluaciones automáticas o manuales por experto.

2.7. Metodología de evaluación recomendada

Es importante tener en cuenta que la revisión mediante herramientas automáticas no es suficiente para asegurar que un sitio web cumpla con las pautas de accesibilidad, ya que estas herramientas solamente comprueban que el código HTML y CSS sea válido y contenga los elementos necesarios para que sea posible cumplir con los criterios de accesibilidad, sin embargo hay muchos aspectos sobre todo semánticos como la no accesibilidad del contenido alternativo que puede pasar inadvertido. Por ejemplo comprobar que el subtítulo de un video corresponda con el contenido original de este, que a día de hoy es imposible interpretar con seguridad mediante software; confirmar que el contraste de colores es el adecuado, que el diseño no pierde precisión al acceder desde distintos navegadores. Por este motivo es necesario que un experto en accesibilidad realice evaluaciones periódicas que aseguren que todo el contenido del sitio cumple con estas pautas. Según las especificaciones del WAI, para realizar la evaluación de un sitio web se debe analizar la sintaxis en todo el sitio web y realizar un análisis sobre una muestra de páginas web mediante dos herramientas automáticas de evaluación de la accesibilidad y mediante una evaluación manual (W3C, 2008).

2.8. Herramientas

En Internet existe un gran número de herramientas que permiten analizar múltiples aspectos de la accesibilidad de una página o sitio web, estas aplicaciones se pueden clasificar en función de su objetivo y funcionalidad en las siguientes (Moreno López, 2010):

- Validadores de accesibilidad, los cuales permiten la revisión automática de páginas web según unas pautas definidas, siendo estas las WCAG, Sección 508 u otras.
- Barras de desarrollo, las cuales permiten ampliar la funcionalidad de un navegador web para permitir a los desarrolladores comprobar el estado de accesibilidad, los errores de gramática del código, la estructura de este u otros elementos.
- Validadores de estándares, que analizan las webs comprobando si presentan errores de (X)HTML o CSS en función de las gramáticas utilizadas.
- Validadores de aspecto, que permiten simular la manera en que las páginas se presentan en distintos navegadores o a personas con determinadas discapacidades.
- Lectores de pantalla, software que permite la utilización del sistema operativo y las distintas aplicaciones mediante el empleo de un sintetizador de voz que "lee y explica" lo que se visualiza en la pantalla, lo que supone una ayuda para las personas con graves problemas de visión o completamente ciegas.
- Magnificadores de pantalla, software o hardware que permite ampliar una región de la pantalla, ayudando a las personas con problemas de visión.
- Navegadores accesibles, que permiten una navegación más fácil para personas sin experiencia o con discapacidades cognitivas o problemas de aprendizaje.
- Navegadores alternativos, navegadores con características especiales, como navegadores solo texto, navegadores que cuentan con magnificadores de pantalla o con sintetizadores de voz, etc.
- Simuladores de personas con discapacidad visual, que permiten simular como visualizaría la web personas con distintas discapacidades visuales.
- Simuladores de personas con discapacidad cognitiva, que ayudan a los desarrolladores a simular este tipo de problema en las páginas web.

- Herramientas orientadas a la epilepsia, que permiten a los desarrolladores identificar que componentes de una página pueden perjudicar a las personas con esta discapacidad.
- Evaluadores de texto, los cuales comprueban la ortografía, la gramática y la dificultad de un texto.
- Evaluadores de color, los cuales comprueban si el contraste del texto es suficiente según las recomendaciones de WCAG.

Utilizando un conjunto de estas herramientas y siguiendo pautas de accesibilidad y usabilidad, se puede conseguir desarrollar sitios web con una calidad muy alta y que cumplan los actuales estándares, permitiendo el acceso a estos a la mayor parte de la población, independientemente de sus características físicas, de acceso, de conexión, etc. Como hemos visto anteriormente la accesibilidad en las webs españolas, ya sea por las obligaciones que impone la legislación actual o por la concienciación de la necesidad de desarrollos accesibles por parte de los diseñadores web o las empresas, está experimentando un importante crecimiento y por ello el uso de estas herramientas de ayuda al desarrollo está cada vez más extendido.

2.8.1. Revisión de la accesibilidad

De cara al mantenimiento de la accesibilidad se hace más necesario contar con herramientas que permitan automatizar lo más posible la realización de análisis de accesibilidad, de forma que con la máxima confiabilidad posible se notifique a las personas responsables de esta accesibilidad cuando se produzca una deficiencia, no teniendo que contar con personas encargadas de esta revisión periódica, lo que en muchos casos conlleva que no se realice o que se haga con una periodicidad excesivamente larga.

Se ha realizado un estudio de los diferentes sistemas que se pueden usar para la gestión y monitorización de la accesibilidad de sitios web que actualmente se encuentran disponibles, existen múltiples y variadas herramientas que permiten la revisión de páginas web o incluso la revisión de sitios web completos. A continuación se realiza una revisión de las más importantes:

- TAW (CTIC, 2010): Esta herramienta permite el análisis de páginas web con respecto a las pautas definidas en WCAG 1.0 y WCAG 2.0, e incluso permite realizar un análisis de un sitio web completo en su versión descargable.
- Hera (Sidar, 2005): Permite el análisis de páginas web unitarias con respecto a las pautas definidas por WCAG 1.0, presentando un interfaz que facilita en gran medida la localización del error.
- TAW Monitor (CTIC, 2010): Permite la monitorización de sitios web en relación a los estándares web a través de su portal, prestando unos servicios básicos de forma gratuita.
- AChecker (ATutor, 2010): Permite el análisis de páginas web unitarias de acuerdo a las pautas definidas por WCAG 1.0 y 2.0, BITV 1.0 (Alemania), Section 508 (EEUU) y Stanca Act (Italia). También permite salvar las revisiones para poder enlazarlas desde el sitio web y personalizar los estándares por los que se desea analizar las páginas. La aplicación se puede utilizar online o descargar para utilizarlo en un servidor propio.

2.9. Deficiencias de las herramientas actuales

El conjunto de herramientas disponibles actualmente permiten el análisis de la accesibilidad de páginas webs individuales, algunas de ellas el análisis además de sitios web completos e incluso existen herramientas para la monitorización continuada de sitios web completos, sin embargo existe una carencia de herramientas que permitan la gestión integral de la accesibilidad de sitios web, permitiendo definir procesos de gestión asociados y permitiendo una gestión más avanzada por un equipo completo de desarrollo, necesidades que pueden aparecer cuando se desea que el equipo de desarrollo web y gestión de la accesibilidad web de una entidad pueda coordinarse mediante una herramienta o en el caso de una empresa dedicada a los desarrollos web que proporcionen servicios de mantenimiento de la accesibilidad a sus clientes. Este aspecto ha motivado la realización del presente trabajo.

2.10. Trabajos relacionados

Previamente a la realización de este trabajo se han analizado en profundidad algunos de los trabajos ya existentes que tienen una importante relación con el actual. A continuación se detallan estos trabajos:

- TAW Monitor (CTIC, 2010): Este servicio web permite la monitorización de uno o varios sitios web en base a análisis automáticos, obteniendo tanto un resultado resumido en base a una métrica que permite una valoración rápida de la situación de la accesibilidad, como un resumen de los problemas más graves y un detalle de estos a nivel de código para ayudar a los desarrolladores. Adicionalmente se ofrecen servicios de notificación de alarmas mediante correo electrónico, así como de los resultados de los análisis.

Debido a que el análisis de sitios web extensos requiere de mucha carga de red y este servicio está hospedado en los servidores de TAW se establecen una serie de limitaciones en el número de dominios monitorizados, así como el número de análisis definidos para estos, el periodo de estos análisis y el número máximo de páginas y de niveles de enlaces recorridos. Estas limitaciones dependen del perfil contratado, adecuado el precio del servicio a las necesidades específicas.

- TAW3 Versión descargable(CTIC, 2010): Esta aplicación multiplataforma desarrollada en Java permite analizar tanto páginas web como sitios web completos sin limitación de enlaces recorridos o páginas web. Se puede configurar para analizar con cualquier nivel de prioridad de WCAG 1.0, así como seleccionar puntos de verificación específicos, por lo que se puede configurar para probar los sitios web en función de la norma UNE, aunque no se proporciona esta opción específica. Este software permite obtener informes en el formato de salida de TAW así como en el formato EARL, definido por W3C.
- Auto Ocawa (Ocawa, 2010): Servicio web que permite, al igual que TAW Monitor, hacer un análisis de la accesibilidad de un sitio web completo (limitado a 100 páginas) y visualizar los resultados de este análisis de forma que se ayude al mantenimiento de la accesibilidad identificando los problemas aparecidos.

2.11. Marco metodológico para accesibilidad

De cara a la implementación de sitios web y aplicaciones accesibles existen pautas y guías que ayudan a la implementación, sin embargo no cubren todo el proceso necesario para un proyecto web, y por lo tanto un marco metodológico que cubra todos los aspectos del desarrollo desde el diseño del sistema hasta el mantenimiento de este sería muy recomendable. A continuación se recoge un marco metodológico orientado en esta dirección:

- AWA, Marco metodológico específico en el dominio de la accesibilidad para el desarrollo de aplicaciones web (Moreno López, 2010): Tesis doctoral que profundiza en los métodos necesarios para desarrollar sitios web accesibles, teniendo en cuenta estos requisitos en todos los ámbitos y desde las primeras fases del desarrollo, de forma que el resultado final presente una alta calidad en relación a la accesibilidad y la usabilidad. Este trabajo ha influido directamente en el presente proyecto, ya que este se deriva de la carencia de muchos sitios web actuales de un sistema eficaz de monitorización y mantenimiento de la accesibilidad, tema que se aborda en dicho trabajo.

2.12. Tecnologías para el desarrollo de aplicaciones web

Previamente al diseño del sistema se ha hecho un estudio de las herramientas del mercado actual orientadas al desarrollo de aplicaciones web en el lenguaje de programación Java, ya que la elección de uno u otro *framework* podrá tener una gran repercusión en el diseño que se deberá hacer del sistema.

De entre todos los marcos de trabajo disponibles se han seleccionado para su estudio los de uso más extendido, los cuales cuentan como norma general con una comunidad de desarrolladores y un soporte mayor, así como una documentación más completa y detallada. A continuación se exponen las conclusiones:

- Struts 2 (The Apache Software Foundation, 2009): Apache Struts 2 es un marco de trabajo de software libre para desarrollo de aplicaciones web que está diseñado para apoyar todo el proceso de desarrollo desde la construcción hasta el despliegue de la aplicación. Struts 2 proporciona respuestas dinámicas en base a la entrada obtenida y la información del sistema y cuenta con un diseño que implementa una arquitectura Modelo Vista Controlador (MVC), permitiendo diseñar aplicaciones con un código limpio. Proporciona también un conjunto de *tags* para realización de llamadas asíncronas al servidor mediante AJAX. Para la captura de los datos de formulario enviados por los usuarios, Struts 2 utiliza JavaBeans, enviando los datos directamente a las clases de acción. Para el desarrollo de aplicaciones en este *framework* se hace uso de los lenguajes de programación Java y JSP y los lenguajes HTML y XML para la configuración del sistema, lo que requiere de ciertos conocimientos en estas tecnologías.

Respecto a las posibilidades de Struts 2 en base a la accesibilidad se destaca que la salida proporcionada por esta herramienta es personalizable mediante plantillas internas que definen el formato de cada *tag*, con la configuración por defecto estas

plantillas producen una salida no accesible, sin embargo se puede usar la plantilla simple, que proporciona una salida mínima, que luego se puede completar con código HTML adicional en las páginas JSP siendo el resultado final accesible. Otro aspecto importante es que las capacidades Ajax para validación de formularios comunicación con el servidor en general, se pueden proporcionar al mismo tiempo mediante Ajax y llamadas convencionales mediante eventos redundantes, mejorando la usabilidad de la aplicación a la vez que se asegura su funcionamiento con navegadores que no dispongan de Javascript o soporte Ajax.

- JavaServer Faces (JSF) (Oracle Corporation, 2010): JavaServer Faces es un *framework* para desarrollo de aplicaciones web que se ha establecido como estándar. El API de JSF hace uso de un diseño basado en componente que facilita el desarrollo de la aplicación mediante herramientas gráficas, de forma que los nodos del sistema y los elementos dentro de estos se pueden crear arrastrando elementos gráficos en un entorno de desarrollo específico. JSF, al igual que Struts 2, proporciona unas librerías de *tags* para el manejo de este marco desde JSP, proporcionando también soporte Ajax mediante estas e implementa una arquitectura MVC. Otro punto importante es que JSF no proporciona una implementación estándar, sino que hay múltiples implementaciones del estándar, cada una de ellas con características propias diferentes.

Respecto a la accesibilidad, la salida que proporciona JSF, en las implementaciones MyFaces (The Apache Software Foundation, 2010) y RichFaces (JBoss Community, 2010) que son las que se han probado, ha presentado un código HTML que incluía gran cantidad de código Javascript y de contenido adicional que no cumplía con los estándares WCAG, además no se ha encontrado una manera de modificar esta salida adecuándola a estos que no requiera de la modificación de código propio de este *framework*.

- Spring MVC (SpringSource, 2010): Spring es un *framework* orientado al desarrollo de proyectos de cualquier envergadura mediante una arquitectura MVC. Spring permite hacer uso de cualquier objeto como comando u objeto de formulario, no siendo necesario implementar interfaces específicos del marco de trabajo. Spring proporciona un sistema de gestión de datos muy flexible.

Respecto a la accesibilidad Spring presenta en su salida estándar ciertos problemas básicos que harían que esta no cumpliera con los estándares, sin embargo estos problemas podrían resolverse realizando configuraciones en esta salida estándar definiendo el formato de la nueva salida.

2.13. Entornos de desarrollo para aplicaciones web

A continuación se estudia el soporte de los principales entornos integrados de desarrollo Java para las tecnologías analizadas anteriormente:

- Eclipse IDE (The Eclipse Foundation, 2010): Entorno de desarrollo de software libre que ofrece una versión orientada a la implementación de aplicaciones web. La comunidad de Eclipse proporciona *plugins* para trabajo con Struts 2 y Spring MVC

ayudando al desarrollo de estos proyectos con arquitectura Modelo Vista Controlador, adicionalmente existen *plugins* para implementación en JSF mediante elementos gráficos.

- NetBeans (Oracle Corporation, 2010): NetBeans proporciona soporte para desarrollar tanto Spring como JSF, incluyendo herramientas gráficas para este último y asimismo existen *plugins* para implementación de aplicaciones Struts 2.

2.14. Conclusiones

Una vez concluido el estudio tecnológico realizado para elegir la tecnología que mejor se adapte a los requisitos de este proyecto se ha tomado la decisión de utilizar el *framework* Struts 2, ya que de esta forma se consigue separar los datos de la aplicación de la interfaz del usuario y de la lógica de control mediante una arquitectura Modelo Vista Controlador (MVC), consiguiendo una gran abstracción y mayor potencia en el desarrollo, así como una estructura firme de funcionamiento sobre la que implementar los aspectos específicos de la aplicación. Además Struts 2 permite que la personalización de la salida de interfaz sea más libre, de esta forma se obtiene un código final que cumpla con los estándares de accesibilidad.

En base al resultado del estudio de los trabajos relativos: funcionalidades que estos ofrecen, carencias que se han encontrado en ellos y necesidades actuales identificadas se han definido unos requisitos. Estos requisitos están recogidos en el Capítulo 3. Para ello se ha tenido en cuenta además, los distintos métodos de evaluación tanto automáticos como manuales de cara a definir los mecanismos que se incluirán en el sistema.

Capítulo 3.

Análisis

Se especifica el análisis de las necesidades del sistema, determinando las capacidades que debe tener y los estándares utilizados así como definiendo los requisitos de usuario y los requisitos de software y detallando la funcionalidad mediante casos de uso:

3.1. Determinación del alcance del sistema

El sistema debe contener una funcionalidad tal que permita a cualquier usuario sin necesidad de que posea conocimientos técnicos de ningún tipo e independientemente de sus capacidades personales y de conexión a internet, así como del dispositivo que utilice para la navegación, la inclusión de incidencias en el sistema.

El objetivo principal de estas incidencias será la notificación de problemas de accesibilidad encontrados durante la navegación en alguno de los sitios web monitorizados por este sistema, esta funcionalidad está encaminada a cumplir con cualquier punto de verificación WCAG del nivel deseado, así como dar un servicio de soporte a los usuarios de las páginas web monitorizadas, asistiéndoles en caso de que encuentren problemas de navegación o cualquier otra índole.

Además de la anterior funcionalidad, el sistema proporcionará un acceso mediante interfaz web a los expertos encargados de la accesibilidad de las páginas web, los cuales podrán ver y editar la información incluida originalmente en las incidencia y crear nuevas incidencias en base al análisis experto de las páginas.

La aplicación contará con un módulo que permitirá consultar los listados de las alertas automáticas para cada uno de los puntos de verificación de las pautas seguidas en el análisis, analizando cada sitio web monitorizado en el sistema por separado.

Un tercer módulo del sistema estará dedicado a la gestión del conocimiento relacionado con la gestión y el mantenimiento de la accesibilidad. Este módulo permitirá consultar y crear instancias de conocimiento recuperables mediante un sistema de clasificaciones múltiples con estructura arbórea.

Asimismo el sistema proporcionará una interfaz de administración que contendrá opciones relativas a la configuración y mantenimiento de la aplicación. Presentando un completo subsistema de manejo de permisos, el cual está basado en el diagrama de la Figura 3-1:

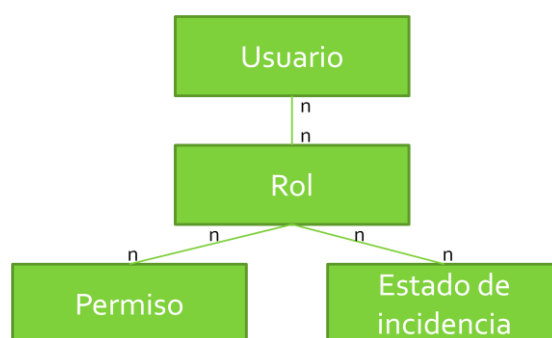


Figura 3-1: Diagrama de permisos y usuarios

Los usuarios están agrupados dentro de los roles que van a desempeñar, siendo estos roles los que presentan un conjunto de permisos y, en caso de que dispongan del permiso para la visualización y edición de incidencias, un conjunto de estados de incidencias que le serían accesibles.

3.1.1. Funcionalidades fuera del alcance del sistema

El sistema no interpretará ni realizará o propondrá acciones en respuesta a las incidencias incluidas, estas acciones deberán ser llevadas a cabo por un experto en accesibilidad, el cual será el responsable de su debido tratamiento.

Asimismo el sistema no está pensado para realizar un mantenimiento automático de los sitios monitorizados, sino una monitorización, la cual ayudará a los responsables del mantenimiento en la realización de sus tareas, así como en la resolución de los problemas aparecidos.

3.1.2. Ventajas de esta herramienta

La actual herramienta no pretende ser un sistema de análisis de páginas o sitios web que sea utilizado en la creación de este, por el contrario pretende prestar una ayuda continua para el mantenimiento de un alto nivel de accesibilidad en un sitio o sitios web, aportando para ello funcionalidades que informen en todo momento del estado de accesibilidad del sitio, así como la posibilidad de integrar los mecanismos de gestión de la accesibilidad de la empresa en la aplicación por medio del sistema de permisos y gestión de estados. Esta aplicación proporciona también un sistema de notificación de incidencias por parte de los usuarios, y de creación de incidencias expertas, las cuales serán enviadas al sistema por especialistas dedicados al análisis de los sitios web, en caso de existir estos.

Este sistema además facilita el cumplimiento del Real Decreto 1494/2007 de 12 de noviembre de 2007, por el que se aprueba el "Reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a las tecnologías, productos y servicios relacionados con la Sociedad de la Información y medios de comunicación social" (BOE, 2007), en el cual se indica que se tiene que proporcionar un sistema de contacto para transmitir las dificultades de acceso, o formular quejas, consultas o sugerencias; mediante los mecanismos de gestión de incidencias, agilizando este proceso y aumentando así la calidad de la atención al usuario.

Asimismo, el sistema proporciona un almacén de conocimiento, el cual se puede utilizar para definir, difundir o coordinar las buenas prácticas seguidas en la empresa, como una base de datos de soluciones a problemas comunes o cualquier otra información de relevancia; estando esta información estructurada según unas clasificaciones arbóreas para un fácil acceso y recuperación.

Otra importante ventaja a tener en cuenta es la facilidad de integración de esta herramienta con la página web que se desee monitorizar, pudiéndose modificar el aspecto de la página de creación de nuevas incidencias con unos simples ajustes en la CSS o la inclusión de código (X)HTML adicional, sin afectar al comportamiento de la aplicación y dando al usuario la impresión de estar dentro del mismo sitio web.

Esta herramienta permite una gran libertad de personalización, adecuándose a las necesidades específicas que tenga cada cliente, permitiendo la modificación de la estructura de permisos en función de la estructura organizativa, los estados de las incidencias en función del proceso de seguimiento que se desee hacer de estas y las clasificaciones, pudiendo personalizarlas según las necesidades del equipo de accesibilidad.

Además el sistema implementa tecnologías de localización e internacionalización, presentando actualmente la interfaz en español e inglés, pero permitiendo incluir nuevos idiomas solamente añadiendo los ficheros de localización correspondientes.

En resumen, este sistema pretende ser más completo que los analizados en el punto 2.10 (Trabajos relacionados), ya que se proporciona de forma integral los servicios de análisis automático, gestión de incidencias de usuarios y almacén de conocimiento. Por otro lado siguiendo un enfoque de diseño centrado en el usuario (DCU), se ha tenido en cuenta opiniones obtenidas de posibles usuarios finales de la aplicación.

3.2. Especificación de estándares y normas

En este punto se especificarán los estándares, leyes, normativas y recomendaciones que se tendrán en cuenta a lo largo del desarrollo del proyecto.

- Para la especificación de los diagramas tanto del análisis como del diseño del sistema de la información, se utilizará el Lenguaje Unificado de Modelado (UML) (Object Management Group, 2009).
- Esta aplicación podrá contener datos personales de los usuarios y por lo tanto se aplicará lo dispuesto en la Ley Orgánica de Protección de Datos (Boletín Oficial del Estado (BOE), 1999).
- En relación con el diseño web se cumplirán en todo caso las especificaciones de WCAG 2.0 con un nivel de prioridad tripe A, recogidas en (W3C, 2008).

3.2.1. Restricciones generales

El sistema debe poder implantarse con facilidad en cualquier sitio web existente, evitando realizar cambios en su código más allá de proporcionar un enlace a la página web de creación de incidencias para los usuarios. El interfaz de gestión de la aplicación permanecerá en cualquier caso separado de la aplicación cliente.

3.2.2. Entorno operacional

La aplicación cliente debe poder utilizarse en cualquier equipo informático, independientemente de la plataforma, hardware y software utilizados, como único requisito será necesario un navegador web y conexión a internet. No se requiere que el navegador web disponga de soporte Javascript, sin embargo es recomendable que el navegador disponga de soporte CSS3 para obtener la mejor presentación, no siendo en ningún caso necesario para hacer uso de la funcionalidad del sistema.

La aplicación servidor deberá poder ejecutarse en cualquier equipo de gama media, sin embargo los requisitos específicos dependerán de la carga de trabajo particular que deba soportar la aplicación.

3.3. Establecimiento de requisitos de usuario

Especifican las necesidades que el sistema debe cubrir con un bajo nivel de detalle y en base a las necesidades del mercado actual, ya que no ha habido ningún cliente entrevistado. Los requisitos se detallan mediante los siguientes atributos:

- **Identificador:** formado por las siglas de Requisito y el tipo del mismo seguido de un número secuencial dado por orden de creación del requisito.
- **Descripción:** explicación formal del propósito del requisito.
- **Necesidad:** valoración de la importancia de cumplir el requisito. La valoración sigue los valores:
 - Alta: el requisito es indispensable, sino se cumplen todos los requisitos de este nivel no se acepta el software bajo ningún concepto.
 - Media: el requisito es muy importante pero si no se cumple un porcentaje mínimo de ellos se puede renegociar el software.
 - Baja: el requisito es importante pero no vital, no cumplir todos los requisitos bajos no implica el rechazo del software pero si una posible renegociación del precio del mismo con el cliente.
- **Prioridad:** si el software se desarrolla en múltiples etapas indica la importancia de cubrir unos requisitos antes de otros
 - Alta: el requisito debe solucionarse en fases iniciales del desarrollo software.
 - Media: se debe solucionar en las fases medias del desarrollo del software.
 - Baja: se soluciona el requisito al final del ciclo de desarrollo.
- **Estabilidad:** mide la posibilidad de cambios en el requisito en el ciclo de vida del proyecto
 - Alta: no va a sufrir cambios.
 - Media: no se esperan cambios o se espera un número de cambios mínimos y no críticos.
 - Baja: es posible que cambien mucho en el tiempo o que el cambio sea crítico en el sistema.
- **Origen:** En el caso de los requisitos de usuario trata de la persona que ha definido el requisito, para el detalle de los requisitos de software trata del requisito de usuario del que proviene.

- **Grado de Verificación, (G. Verificación):** mide si se puede comprobar el cumplimiento del requisito.
 - Alta: el requisito es fácil de verificar.
 - Media: el requisito se puede verificar pero las pruebas que implica son numerosas.
 - Baja: el número de pruebas que implica la Verificación del requisito es demasiado elevado o el cumplimiento del requisito es subjetivo de los usuarios.

Tabla 3-1: RU01

Identificador: RU01	
Descripción: El sistema debe ser accesible y podrá ser utilizado por cualquier usuario capaz de manejar un navegador web.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Tutor	G. Verificación: Alta

Tabla 3-2: RU02

Identificador: RU02	
Descripción: La aplicación cliente debe poder utilizarse en cualquier equipo informático, independientemente de la plataforma, hardware y software utilizados, como único requisito será necesario un navegador web que cumplan con los estándares actuales sobre XHTML y CSS, así como conexión a Internet.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: Tutor	G. Verificación: Media

Tabla 3-3: RU03

Identificador: RU03	
Descripción: Las incidencias deben almacenarse en la aplicación y estar disponibles para ser consultadas por los técnicos en un tiempo inferior a un minuto.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

Tabla 3-4: RU04

Identificador: RU04	
Descripción: La aplicación debe proporcionar un sistema de clasificación de incidencias que permita especificar de manera efectiva: <ul style="list-style-type: none"> • Estado de la incidencia • Importancia de la incidencia 	
Necesidad: Media	
Prioridad: Media	Estabilidad: Media
Origen: Proyectando	G. Verificación: Alta

Tabla 3-5: RUo5

Identificador: RUo5	
Descripción: En caso de fallo en cualquiera de los sistemas los datos deberán permanecer inalterados.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Media

Tabla 3-6: RUo6

Identificador: RUo6	
Descripción: Se deben proporcionar mecanismos para gestión de usuarios, roles y permisos de roles, limitando el acceso a las distintas partes de la aplicación en función de estos permisos.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

Tabla 3-7: RUo7

Identificador: RUo7	
Descripción: Las incidencias pueden mostrarse a los usuarios con distintos roles en función de su estado.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

Tabla 3-8: RUo8

Identificador: RUo8	
Descripción: Se debe permitir a los administradores gestionar las clasificaciones de las incidencias.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

Tabla 3-9: RUo9

Identificador: RUo9	
Descripción: Se debe permitir a los administradores gestionar los estados de las incidencias.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

Tabla 3-10: RU10

Identificador: RU10	
Descripción: Se debe permitir a los administradores gestionar los sitios monitorizados.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

Tabla 3-11: RU11

Identificador: RU11	
Descripción: Se debe permitir ver, editar y eliminar las incidencias.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

Tabla 3-12: RU12

Identificador: RU12	
Descripción: Se debe permitir ver el listado de las incidencias a las que tiene acceso el usuario.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

Tabla 3-13: RU13

Identificador: RU13	
Descripción: Se debe evitar que la aplicación se quede sin usuarios administradores, perdiendo la capacidad de reconfigurarla.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

Tabla 3-14: RU14

Identificador: RU14	
Descripción: Se debe proporcionar un mecanismo para almacenar y recuperar instancias de conocimiento sobre accesibilidad, siendo estos clasificables.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Media
Origen: Tutor	G. Verificación: Alta

Tabla 3-15: RU15

Identificador: RU15	
Descripción: El sistema debe ser capaz de analizar los sitios web monitorizados de forma automática, identificado alertas de posibles problemas de accesibilidad, tal y como se puede hacer a través de las herramientas Hera ¹ o TAW ² , mostrando los resultados a los usuarios autorizados.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

Tabla 3-16: RU16

Identificador: RU16	
Descripción: El sistema debe estar disponible íntegramente en castellano y en inglés y se tiene que poder ampliar a otros idiomas fácilmente.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

3.4. Especificación de casos de uso

A continuación se muestran los casos de uso en función del tipo de acceso y de los permisos de los que disponga el usuario:

3.4.1. Caso de uso, inclusión de incidencias desde el formulario web

El usuario, al acceder al formulario web podrá rellenar éste y enviar los datos introducidos en el formulario al servidor para la creación de la nueva incidencia, así borrar los datos ya introducidos en el formulario de forma que quede vacío de nuevo. El *feedback* hacia el cliente proporcionado en esta acción será un mensaje que le indique que la incidencia se ha creado correctamente, en caso de ser conveniente proporcionar más información al usuario se hará mediante correo electrónico en caso de que se haya especificado y quedará a cargo de los responsables de la gestión de las incidencias.

¹ Hera. Revisando la Accesibilidad con Estilo. <http://www.sidar.org/hera/>

² TAW. Test de Accesibilidad Web. <http://www.tawdis.net/>

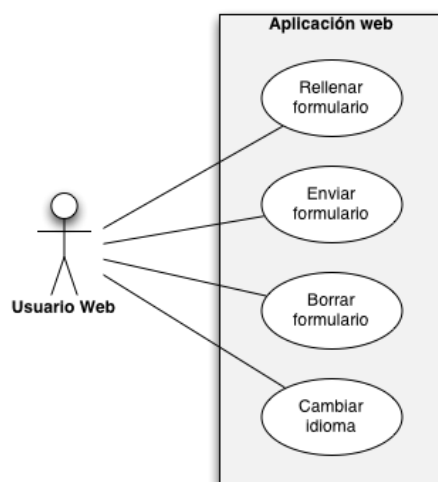


Figura 3-2: Diagrama de casos de uso, inclusión de incidencias desde el formulario web

3.4.2. Caso de uso, interfaz de gestión, usuario no autenticado

El usuario no autenticado solo puede realizar las acciones de autenticarse en la aplicación y de cambiar el idioma del interfaz.

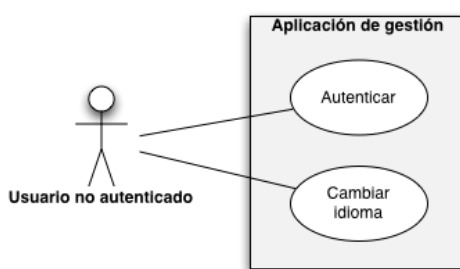


Figura 3-3: Diagrama de casos de uso, usuario no autenticado

3.4.3. Casos de uso, interfaz de gestión, usuario con permisos de visualización de incidencias

Los usuarios con permisos de visualización de incidencias podrán realizar las siguientes acciones, las cuales están relacionadas con la visualización y edición de las incidencias existentes:

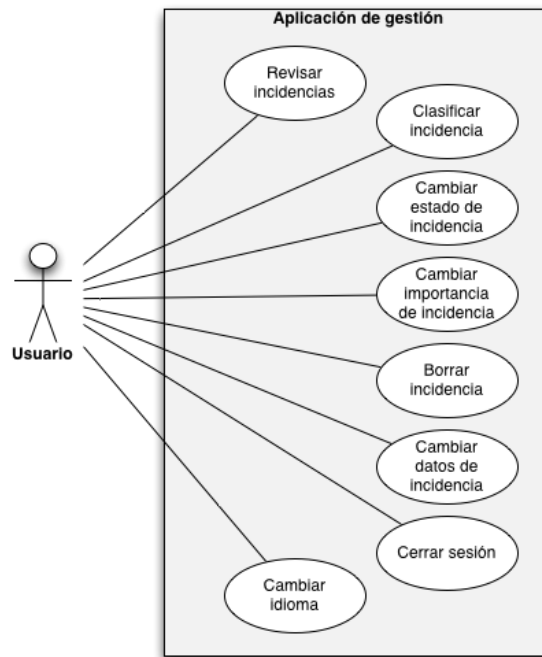


Figura 3-4: Diagrama de casos de uso servidor, usuario con permisos de visualización de incidencias

3.4.4. Casos de uso, interfaz de gestión, usuario con permisos de creación de incidencias

Los usuarios con permisos de creación de incidencias podrán realizar las siguientes acciones, las cuales están relacionadas con la generación de nuevas incidencias:

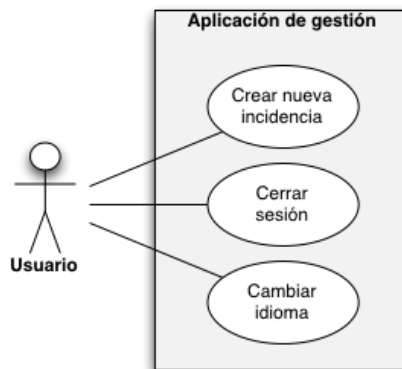


Figura 3-5: Diagrama de casos de uso servidor, usuario con permisos de creación de incidencias

3.4.5. Casos de uso, interfaz de gestión, usuario con permisos de visualización de alertas automáticas

Los usuarios con permisos de visualización de alertas automáticas podrán realizar las siguientes acciones:

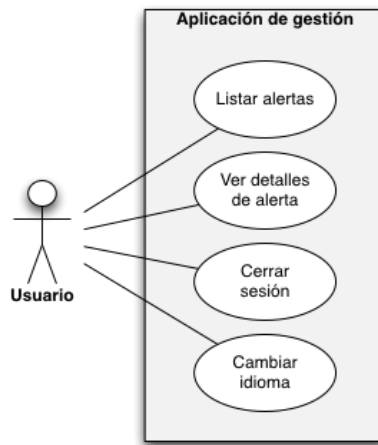


Figura 3-6: Diagrama de casos de uso servidor, usuario con permisos de creación de incidencias

3.4.6. Casos de uso, interfaz de gestión, usuario con permisos de visualización de conocimiento

Los usuarios con permisos de visualización de conocimiento podrán realizar las siguientes acciones:

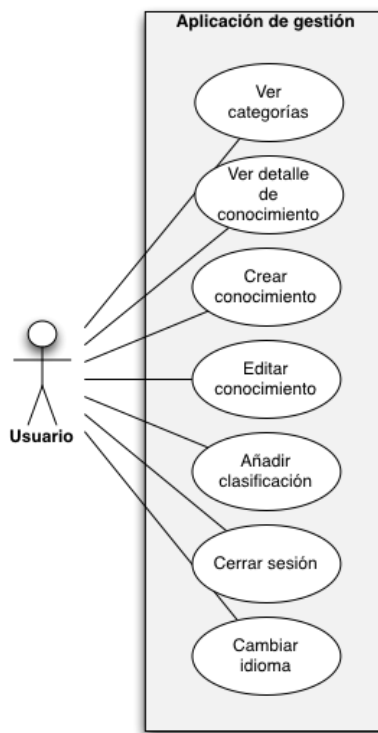


Figura 3-7: Diagrama de casos de uso servidor, usuario con permisos de creación de incidencias

3.4.7. Casos de uso, interfaz de gestión, usuario con permisos de administración

El rol administrador permite realizar tareas de mantenimiento y configuración sobre la aplicación y el resto de usuarios, roles y demás parámetros.

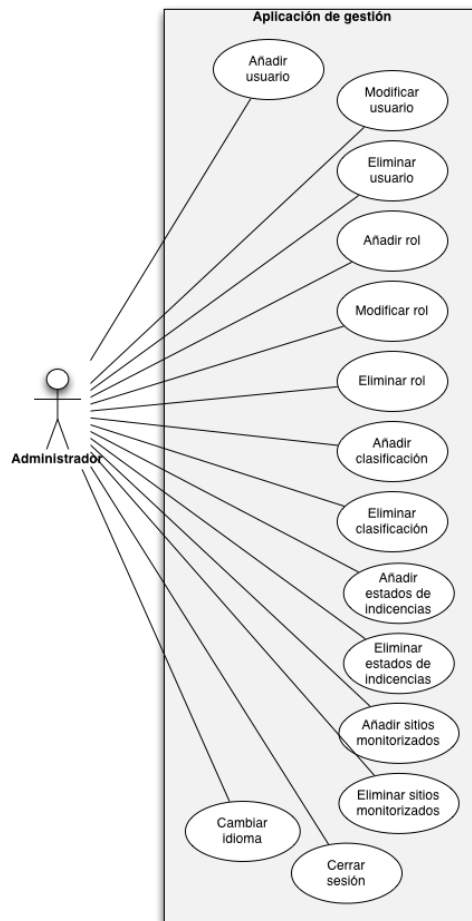


Figura 3-8: Diagrama de clases de uso servidor, rol administrador

3.5. Establecimiento de requisitos software

Los requisitos software detallan la especificación completa del sistema a desarrollar, precisando en su funcionalidad y en sus restricciones de diseño y funcionamiento.

3.5.1. Requisitos funcionales

Tabla 3-17: RFunco1

Identificador: RFunco1	
Descripción: Se presentarán las incidencias a los usuarios que las consulten según los siguientes criterios: <ul style="list-style-type: none"> • Estado de la incidencia, mostrándose al usuario las incidencias que le permita su rol. • Importancia de la incidencia, la cual se indicará de forma clara para que estas puedan ser atendidas en el orden más conveniente. 	
Necesidad: Media	
Prioridad: Media	Estabilidad: Media
Origen: RUo4	G. Verificación: Alta

Tabla 3-18: RFunco2

Identificador: RFunco2	
Descripción: El administrador podrá crear y eliminar usuarios así como modificar los datos y los roles asociados a los existentes.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: RUo6	G. Verificación: Alta

Tabla 3-19: RFunco3

Identificador: RFunco3	
Descripción: El administrador podrá crear y eliminar roles así como modificar los datos y los permisos asociados a los existentes.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: RUo6	G. Verificación: Alta

Tabla 3-20: RFunco4

Identificador: RFunco4	
Descripción: El administrador podrá crear y eliminar los estados de las incidencias.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: RU11	G. Verificación: Alta

Tabla 3-21: RFunco5

Identificador: RFunco5	
Descripción: El administrador podrá determinar que roles tendrán acceso a las incidencias que estén en cada uno de los estados, las incidencias se mostrarán a los usuarios en función de estos estados.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: RUo7	G. Verificación: Alta

Tabla 3-22: RFunco6

Identificador: RFunco6	
Descripción: Cada usuario podrá listar y visualizar las incidencias que hayan sido asignadas a su rol en función de su estado.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: RUo7	G. Verificación: Alta

Tabla 3-23: RFunco7

Identificador: RFunco7	
Descripción: El administrador podrá añadir y eliminar clasificaciones de las incidencias.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: RUo8	G. Verificación: Alta

Tabla 3-24: RFunco8

Identificador: RFunco8	
Descripción: El administrador podrá añadir y eliminar estados de las incidencias.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: RUo9	G. Verificación: Alta

Tabla 3-25: RFuncog

Identificador: RFuncog	
Descripción: El administrador podrá añadir y eliminar los sitios monitorizados. Al añadir un nuevo sitio se iniciará automáticamente su análisis. Al eliminarlos se eliminará de la base de datos todas las entradas relacionadas con este.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: RU10	G. Verificación: Alta

Tabla 3-26: Rfunc10

Identificador: Rfunc10	
Descripción: Se debe permitir ver, editar y eliminar las incidencias para las que tenga permiso el usuario.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: RU11	G. Verificación: Alta

Tabla 3-27: Rfunc11

Identificador: Rfunc11	
Descripción: Se debe permitir ver el listado de las incidencias a las que tiene acceso el usuario, mostrándole los datos más relevantes.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: RU12	G. Verificación: Alta

Tabla 3-28: Rfunc12

Identificador: Rfunc12	
Descripción: Se debe permitir ver las instancias de conocimiento a través de una clasificación de estos a los usuarios que tengan permiso para ello.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Media
Origen: RU14	G. Verificación: Alta

Tabla 3-29: Rfunc13

Identificador: Rfunc13	
Descripción: Se debe permitir crear nuevas instancias de conocimiento y almacenarlas según la clasificaciones. Cada artículo puede tener varias clasificaciones.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Media
Origen: RU14	G. Verificación: Alta

Tabla 3-30: Rfunc14

Identificador: Rfunc14	
Descripción: El sistema utilizará un servicio web externo, escrito en PHP y adaptado desde el código de Hera. Este servicio proporcionará la misma salida que la aplicación Hera original, pero en un formato XML en lugar del HTML original, de forma que el intercambio de datos sea más efectivo y fiable.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: RU15	G. Verificación: Alta

Requisitos de desempeño

Tabla 3-31: RDesempo1

Identificador: RDesempo1	
Descripción: El tiempo desde el envío de la incidencia por parte del usuario hasta el almacenamiento en la base de datos para su posterior consulta será siempre inferior a 1 minuto, salvo error del sistema y el tiempo de espera de un usuario que incluya una nueva incidencia será siempre inferior a 10 segundos desde el envío del formulario hasta el mostrado del mensaje de recepción correcta con un equipo y una conexión a Internet estándar.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: RU03	G. Verificación: Alta

3.5.2. Requisitos de interfaz

Tabla 3-32: RInterfo1

Identificador: RInterfo1	
Descripción: El sistema debe cumplir con las pautas definidas por WCAG con al menos un nivel doble A.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: RUo1	G. Verificación: Alta

Tabla 3-33: RInterfo2

Identificador: RInterfo2	
Descripción: El sistema hará uso de funciones de localización e internacionalización, proporcionando la posibilidad de visualizar el interfaz tanto en castellano como en inglés.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: RU16	G. Verificación: Alta

Tabla 3-34: RInterfo3

Identificador: RInterfo3	
Descripción: El coste de la ampliación del sistema a un nuevo lenguaje será mínimo, necesitando solamente de la definición de un nuevo fichero de traducciones.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: RU16	G. Verificación: Alta

3.5.3. Requisitos operacionales

Tabla 3-35: ROpero1

Identificador: ROpero1	
Descripción: En el caso de que la base de datos no esté disponible o no haya sido configurada correctamente se mostrará un error al usuario.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Media

3.5.4. Requisitos de verificación

Tabla 3-36: RVerifo1

Identificador: RVerifo1	
Descripción: La aplicación cliente debe poder utilizarse haciendo uso de cualquier navegador mayoritario actualizado.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: RUo2	G. Verificación: Media

Tabla 3-37: RVerifo2

Identificador: RVerifo2	
Descripción: No se hará uso de scripts del lado del cliente, maximizando la compatibilidad con navegadores y agentes de usuario.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: RU02	G. Verificación: Media

Tabla 3-38: RVerifo2

Identificador: RVerifo3	
Descripción: Se cumplirá con los estándares actuales XHTML(W3C, 2000) y CSS(W3C, 2008) definidos por el W3C.	
Necesidad: Media	
Prioridad: Media	Estabilidad: Alta
Origen: RU02	G. Verificación: Media

3.5.5. Requisitos de documentación

Tabla 3-39: RDocuo1

Identificador: RDocuo1	
Descripción: El sistema contará con un manual de usuario accesible desde la interfaz, el cual deberá detallar la funcionalidad completa de este.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Tutor	G. Verificación: Alta

3.5.6. Requisitos de seguridad

Tabla 3-40: RSeguo1

Identificador: RSeguo1	
Descripción: En caso de fallo en cualquiera de los sistemas los datos deberán permanecer inalterados y la base de datos en un estado consistente.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: RU05	G. Verificación: Media

Tabla 3-41: RSeguo2

Identificador: RSeguo2	
Descripción: Se debe impedir eliminar al último administrador para evitar entrar en un estado no recuperable.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: RU13	G. Verificación: Alta

3.5.7. Requisitos de calidad

Tabla 3-42: RCalo1

Identificador: RCalo1	
Descripción: El sistema debe ser tolerante a fallos, así si se produce una entrada errónea por parte del usuario, se le debe notificar para que corrija el error y en ningún caso generar un error de sistema.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

3.5.8. Requisitos de entrega

Tabla 3-43: REntro1

Identificador: REntro1	
Descripción: El proyecto en el momento de su entrega debe presentar tanto el código de aplicación, el cual debe cumplir con los requisitos definidos, así como la documentación contenida en el presente documento, donde se recoge todo lo concerniente al proyecto de fin de carrera.	
Necesidad: Alta	
Prioridad: Alta	Estabilidad: Alta
Origen: Proyectando	G. Verificación: Alta

3.6. Matriz de trazabilidad de requisitos de software frente a requisitos de usuario

Tabla 3-44: Matriz de trazabilidad requisitos software – requisitos de usuario

Requisitos software	Requisitos de usuario															
	RUo1	RUo2	RUo3	RUo4	RUo5	RUo6	RUo7	RUo8	RUo9	RU10	RU11	RU12	RU13	RU14	RU15	RU16
Rfunc01				X												
Rfunc02						X										
Rfunc03						X										
Rfunc04											X					
Rfunc05							X									
Rfunc06							X									
Rfunc07								X								
Rfunc08									X							
Rfunc09										X						
Rfunc10											X					
Rfunc11												X				
Rfunc12														X		
Rfunc13														X		
Rfunc14															X	
RDesemp01			X													
RInterfo1	X															

RInterfo2																X
RInterfo3																X
ROper01																
RVerifo1		X														
RVerifo2		X														
RVerifo3		X														
RDocu01																
RSequ01					X											
RSequ02												X				
RCalo1																
REntro1																

3.7. Diagramas de estado

Se muestran los diagramas de estado para algunas de las operaciones que permite la aplicación de forma que se detalla el funcionamiento de esta.

3.7.1. Envío de formulario web

En la Figura 3-9 se muestran un diagrama de los estados por los que pasa la aplicación cuando se añade una incidencia a través del formulario:

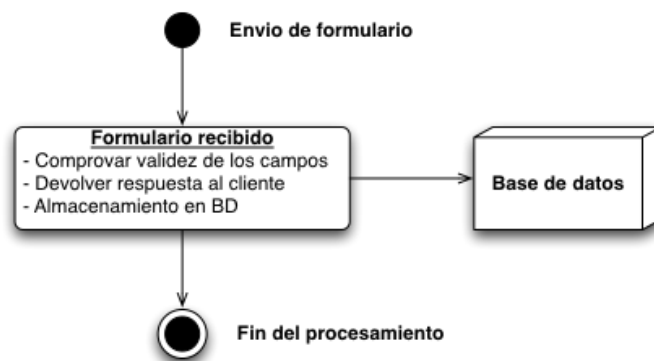


Figura 3-9: Diagrama de estados, inserción de incidencia

Los pasos que se deben realizar son comprobar que los campos requeridos están presentes y que el formato de todos ellos es correcto, almacenar la nueva incidencia en la base de datos y enviar la respuesta al usuario indicando que la incidencia ha sido añadida.

3.7.2. Revisión de incidencias

En la Figura 3-10 se muestra un diagrama de los estados por los que pasa la aplicación de forma genérica al realizar las operaciones más comunes en el transcurso de la vida de una incidencia en el sistema:

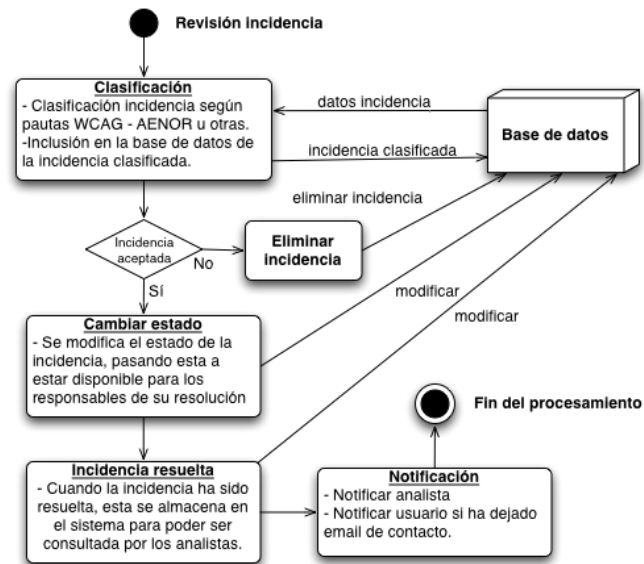


Figura 3-10: Diagrama de estados, revisión de incidencias

Capítulo 4.

Diseño

Se especifica el diseño de la aplicación, incluyendo arquitectura, clases y diseño de la base de datos, así como el diseño detallado:

4.1. Tecnologías utilizadas

Ya que la presente aplicación requiere múltiples servicios dinámicos por parte del servidor, incluyendo persistencia de datos y manejo de distinta presentación en función de los usuarios y de sus roles, se ha seleccionado cuidadosamente un conjunto de tecnologías que presenten ventajas para la implementación de aplicaciones complejas. Finalmente, en base a los conocimientos actuales del programador y las ventajas de cada tecnología en particular se han elegido las siguientes:

- La programación de la aplicación del lado servidor se realizará en el lenguaje Java en su versión 1.6. De esta forma se pueden aprovechar las capacidades que ofrece la tecnología JSP (Java Server Pages) que permite generar código dinámico para web al mismo tiempo que permite el acceso al código Java mediante scripts incluidos dentro del código HTML y que es interpretado en el servidor antes de enviar la página web con la información específica.
- Ya que se pretende separar los datos de la aplicación de la interfaz del usuario y de la lógica de control mediante una arquitectura Modelo Vista Controlador (MVC), se ha hecho uso del *framework* de software libre Struts 2 (The Apache Software Foundation, 2009), debido al uso de este *framework* se consigue una gran abstracción y mayor potencia en el desarrollo, ya que proporciona una estructura firme de funcionamiento sobre la que implementar los aspectos específicos de la aplicación.
- De cara a la persistencia de los datos del sistema se ha elegido utilizar una base de datos MySQL, ya que esta cubre ampliamente las necesidades con respecto a esta herramienta, al mismo tiempo que proporciona numerosas ventajas debido a su facilidad de portabilidad e instalación con respecto a otros gestores de bases de datos. Para el manejo del sistema de base de datos se hace uso del *framework* Hibernate (Red Hat Middleware, LLC, 2010), que proporciona las herramientas necesarias para realizar un mapeo objeto-relacional. De esta forma se facilita el mapeo de atributos entre la base de datos relacional utilizada y el modelo de objetos de la aplicación. La especificación de los parámetros se realiza en ficheros XML específicos.

- El analizador de accesibilidad que utiliza la aplicación para la obtención de las alertas automáticas está basado en el analizador Hera³ implementado en PHP y por lo tanto las modificaciones realizadas sobre este software se han realizado en ese lenguaje. Se ha elegido hacer uso de este analizador sobre otros como AChecker⁴ que ya ofrecían una salida en formato XML debido a que la cantidad de información aportada es significativamente mayor, permitiendo que las alertas automáticas sean mucho más útiles al usuario.

4.1.1. Justificación

Se ha elegido el lenguaje de programación Java para el desarrollo de la aplicación debido en parte a la extensión actual de este lenguaje en el desarrollo de aplicaciones web corporativas, para lo cual presenta un marco de trabajo especializado, J2EE; y debido también a la experiencia ya adquirida de programación en este lenguaje, lo que ha facilitando tanto el diseño como la implementación de la aplicación.

La elección de Struts sobre otras soluciones similares disponibles en el conjunto de tecnologías actuales como implementaciones de JSF (Oracle Corporation, 2010) entre las que se pueden destacar MyFaces (The Apache Software Foundation, 2010) o RichFaces (JBoss Community, 2010); o *frameworks* como Spring(SpringSource, 2010), se ha realizado en base a que la personalización de la salida de interfaz era más libre, permitiendo adecuarla a una salida que cumpliera con los estándares de accesibilidad necesarios. En un principio se buscó un marco de trabajo que permitiera hacer interfaces accesibles de forma inmediata, pero en ninguno de los casos se consiguió y la opción elegida fue hacer uso de Struts 2, limitando la generación de elementos HTML por sus plantillas internas al mínimo e incluyendo estos elementos en las páginas JSP de forma correcta, asegurando de este modo que la estructura y contenido del documento cumplía con los estándares tanto relativos a HTML y CSS como a la accesibilidad web.

La elección del uso de Hibernate sobre el manejo directo de la base de datos a través del conector ha sido debido a las preferencias del cliente y también en parte una decisión de carácter estético y por motivos de aprendizaje, ya que la complejidad de la base de datos no habría supuesto complicaciones en la implementación de las consultas y el manejo de los elementos persistentes.

4.2. Arquitectura

El sistema sigue una arquitectura de alto nivel cliente servidor tanto para el módulo web de creación de incidencias como para el portal de gestión, el cual sigue el diagrama de la Figura 4-1:

³ Hera. Revisando la Accesibilidad con Estilo. <http://www.sidar.org/hera/>

⁴ Dirección web: <http://www.atutor.ca/achecker/>

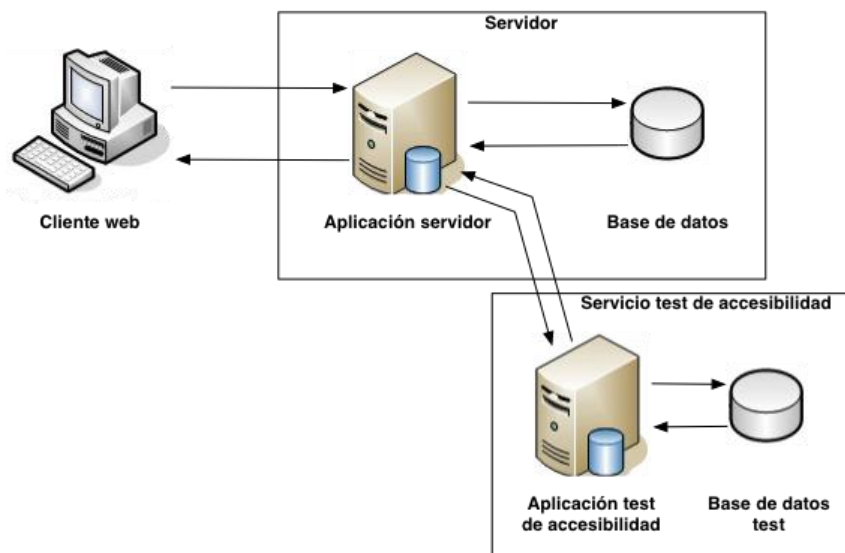


Figura 4-1: Arquitectura del sistema

Esta arquitectura es la utilizada hasta la actualidad para el desarrollo de aplicaciones web y se basa en un esquema en el que el cliente realiza peticiones al servidor y este le proporciona una respuesta en base a estas. La principal ventaja de esta arquitectura recae en la centralización de la gestión de la información y la separación de responsabilidades, lo que permite simplificar el diseño de la aplicación. En este caso, el usuario interactúa directamente con el cliente y este a su vez envía la petición al servidor el cual la procesa y responde. Esta arquitectura presenta sin embargo ciertas limitaciones, como la imposibilidad de realizar comunicaciones en sentido inverso, y que por lo tanto el servidor no sea capaz de enviar al cliente notificaciones o cualquier otro tipo de información, ya que en ningún caso se iniciará una comunicación desde el lado servidor. Esta limitación se ha de tener en cuenta desde el diseño de la aplicación, ya que puede provocar complicaciones e incluso la imposibilidad de implementar ciertas funcionalidades, siendo necesario buscar alternativas que permitieran este tipo de comunicación inversa de manera real o virtual.

Adquiriendo esta arquitectura se pretende maximizar el volumen de trabajo que puede procesar la aplicación y la escalabilidad a largo plazo. Asimismo se mantiene la orientación habitual de los procesos web, los cuales suelen usar esta arquitectura.

El servicio de test de accesibilidad es un servicio web, basado en el test de accesibilidad Hera⁵ y modificado para proporcionar una salida en XML, la cual es más adecuada para el posterior tratamiento de los datos. Este servicio será accedido desde el servidor de forma asíncrona e independientemente de las peticiones que se hagan a este, de forma que los datos se actualizan y se almacenan en la base de datos para que estén disponibles para futuras consultas.

⁵ Hera. Revisando la Accesibilidad con Estilo. <http://www.sidar.org/hera/>

4.2.1. Arquitectura específica de la aplicación servidor

La arquitectura utilizada en el sistema servidor corresponde a un patrón Modelo Vista Controlador (MVC), de esta forma se consigue la separación de los datos, la lógica de control y el interfaz de la aplicación.

Con el objetivo de facilitar la implementación de esta arquitectura se hace uso del *framework* Struts 2 (The Apache Software Foundation, 2009), el cual proporciona los mecanismos necesarios para la realización de una aplicación con arquitectura MVC, abstrayendo al programador de la codificación a este nivel. De esta manera la aplicación final será significativamente más estable y optimizada, al mismo tiempo que una menor complejidad del código y mayor modularidad favorecerá la posterior ampliación o modificación de funcionalidades.

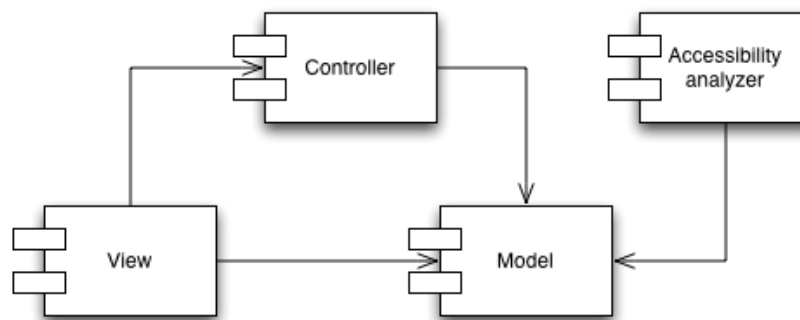


Figura 4-2: Arquitectura del servidor

- **Modelo (Model):** Almacena toda la información del sistema de manera persistente asegurando la integridad de todos los datos. Se hace uso de un sistema de bases de datos, gestionado a través de Hibernate, lo que hace que la aplicación sea independiente del gestor de base de datos, pudiendo cambiarse sin alterar el resto del sistema y mapeando los datos de las tablas a objetos Java de manera transparente a la aplicación. Además dentro del componente Modelo se engloban los JavaBeans utilizados para la persistencia del lado de la aplicación, y que Hibernate se encarga de sincronizar con la base de datos.
- **Vista (View):** Proporciona, mediante un interfaz web los métodos necesarios para que los usuarios puedan acceder a la aplicación. Este interfaz está basado en un conjunto de componentes JSP y Struts, que contienen tanto la estructura estática de las webs XHTML resultantes como la lógica necesaria para integrar el contenido dinámico obtenido de la base de datos dentro de estas páginas. Estas páginas web XHTML serán interpretadas y renderizadas por el navegador web del usuario.
- **Controlador (Controller):** Proporciona la lógica del sistema, recibiendo las entradas desde la vista en forma de peticiones "HTTPRequest" desde los navegadores web y manejando los eventos, encargándose de invocar cambios en el modelo y de enviar las respuestas apropiadas a la vista. Permite la ejecución de los métodos necesarios para todo el manejo del sistema y es el responsable de la integridad de los datos de la aplicación. El controlador está formado por las clases de acción y los interceptores de Struts así como un conjunto de clases Java de apoyo.

- Analizador de accesibilidad (Accessibility analyzer): Analiza periódicamente los sitios web que están incluidos en la base datos mediante el servicio de análisis externo, recorriendo para ello todas las páginas de estos sitios web, las cuales serán analizadas por separado. La información obtenida se almacena en el modelo (base de datos) como incidencias automáticas, las cuales se podrán consultar posteriormente para mostrar los datos al usuario.

4.3. Pautas de diseño

En el desarrollo de la aplicación se han seguido estrictamente las recomendaciones de la Programación Orientada a Objetos (POO), en particular los conceptos de encapsulación y polimorfismo. Para ello toda la aplicación se ha desarrollado dentro del paquete `"com.jar.pfc"`, y se ha reducido la visibilidad de las clases y métodos al mínimo, evitando que estas sean visibles desde fuera del paquete en la medida de lo posible. Esto, aunque para la actual aplicación no es una ventaja notable, ya que no se pretende que sea utilizada como librería desde otra aplicación, es una práctica muy recomendable, ya que el uso de clases, métodos o atributos `"public"` o `"protected"` permiten su uso desde código externo, y por lo tanto estos quedarían añadidos al API público de nuestra aplicación, solo el código `"private"` o `"package-protected"` evita la inclusión de los elementos en el API público.

Estas prácticas chocan con la costumbre extendida de utilizar los paquetes como una herramienta para organizar las clases, permitiendo hacerlo en una estructura similar a la de carpetas, pero como se ha indicado anteriormente, desde el punto de vista de la orientación a objetos, estas prácticas están completamente desaconsejadas al romper con la encapsulación proporcionada por el sistema de paquetes. Cada uno de los paquetes engloba un conjunto de elementos de código que presenta un interfaz público para la interacción con código de otros paquetes.

4.3.1. Problemas encontrados en la aplicación de estas pautas

Tanto Hibernate como Struts necesitan hacer uso de los JavaBeans, y por lo tanto estos deben de ser clases públicas y sus métodos `get` y `set` tienen que ser también públicos para poder estar accesibles desde fuera del paquete para los objetos de estos *framework*.

Asimismo, las clases que implementan las acciones de Struts (implementan `"ActionSupport"`) deben ser invocadas desde este *framework*, y por lo tanto también necesitan ser declaradas como `"public"` obteniendo visibilidad externa.

4.4. Diseño detallado

A continuación se especifica el diseño seguido en la aplicación, especificando un nivel de detalle mayor. La Figura 4-3 muestra una vista global del sistema incluyendo las relaciones entre las clases, a continuación se detallan los módulos de la aplicación y se especifican los diagramas de clases para cada uno de estos con un nivel de detalle mayor.

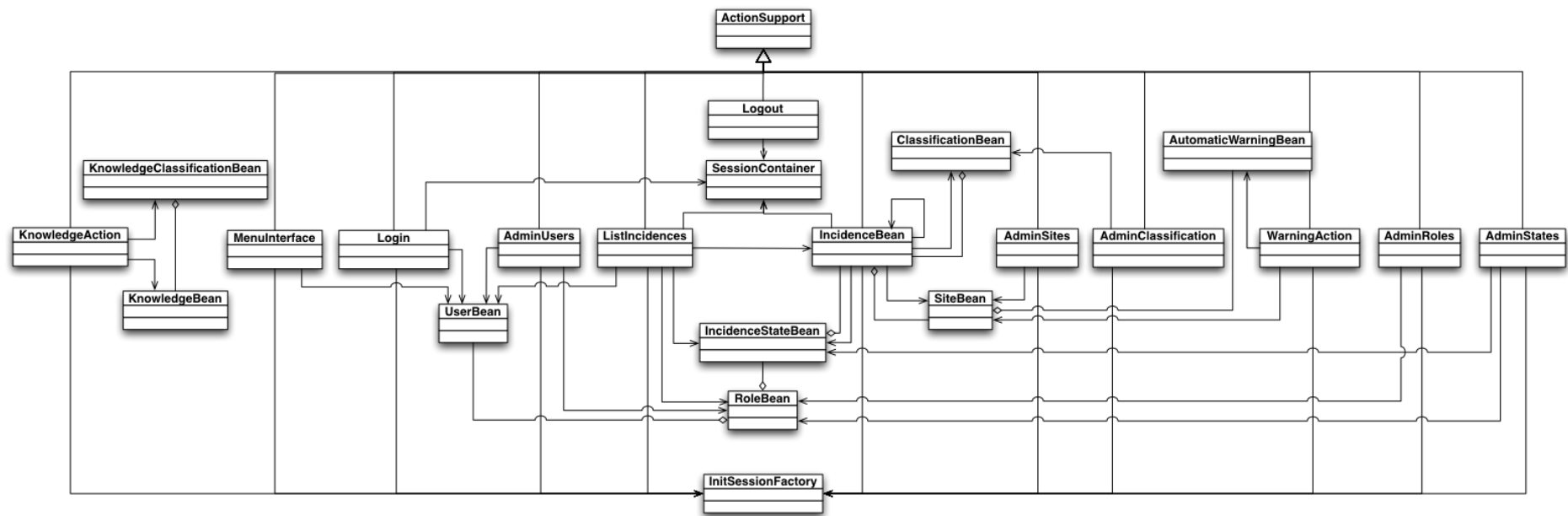


Figura 4-3: Diagrama de clases, vista global

4.4.1. Módulos del sistema

El sistema ha sido diseñado de una forma modular, de esta manera se ha conseguido una notable diferenciación e independencia entre las diferentes partes de la aplicación y por lo tanto mayor facilidad para modificar o eliminar alguno de los módulos existentes, así como añadir otros nuevos. La figura Figura 4-4 muestra los módulos existentes y sus relaciones:

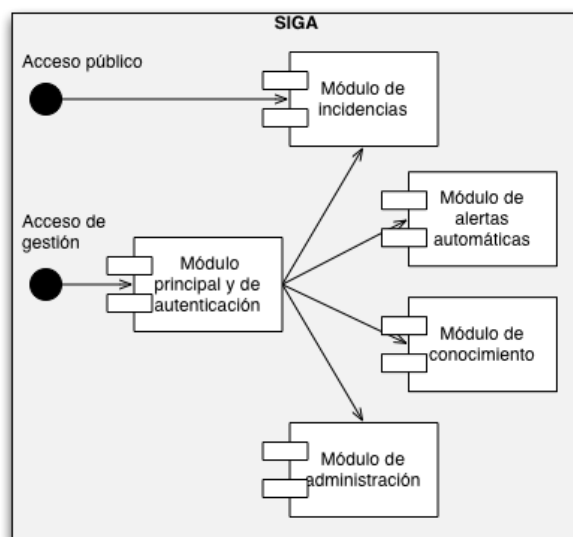


Figura 4-4: Módulos del sistema

- El módulo principal y de autenticación permite el acceso a los usuarios gestores al sistema, y el acceso desde ahí a cualquiera de los otros módulos incluidos.
- El módulo de incidencias permite la gestión de las incidencias existentes en el sistema, la inclusión de nuevas incidencias por parte de usuarios expertos y la creación de nuevas incidencias por usuarios anónimos mediante un acceso público.
- El módulo de alertas automáticas tiene dos funcionalidades, la primera de ellas es explorar los sitios web configurados obteniendo de ellos las alertas de accesibilidad y almacenándolas en la base de datos; la segunda funcionalidad consiste en permitir el acceso a los usuarios a esta información mediante su interfaz.
- El módulo de conocimiento permite la creación, clasificación y consulta de instancias de conocimiento.
- El módulo de administración permite la gestión de las configuraciones del sistema.

En el apartado 4.4.2 Detalle de los módulos se puede apreciar un mayor detalle para cada uno de los módulos.

4.4.2. Detalle de los módulos

A continuación se detallan los diagramas de clases separados por módulos, de forma que estos transmitan la información de manera clara y sencilla.

Módulo principal y de autenticación

El módulo principal y de autenticación presenta el siguiente digrama de clases:

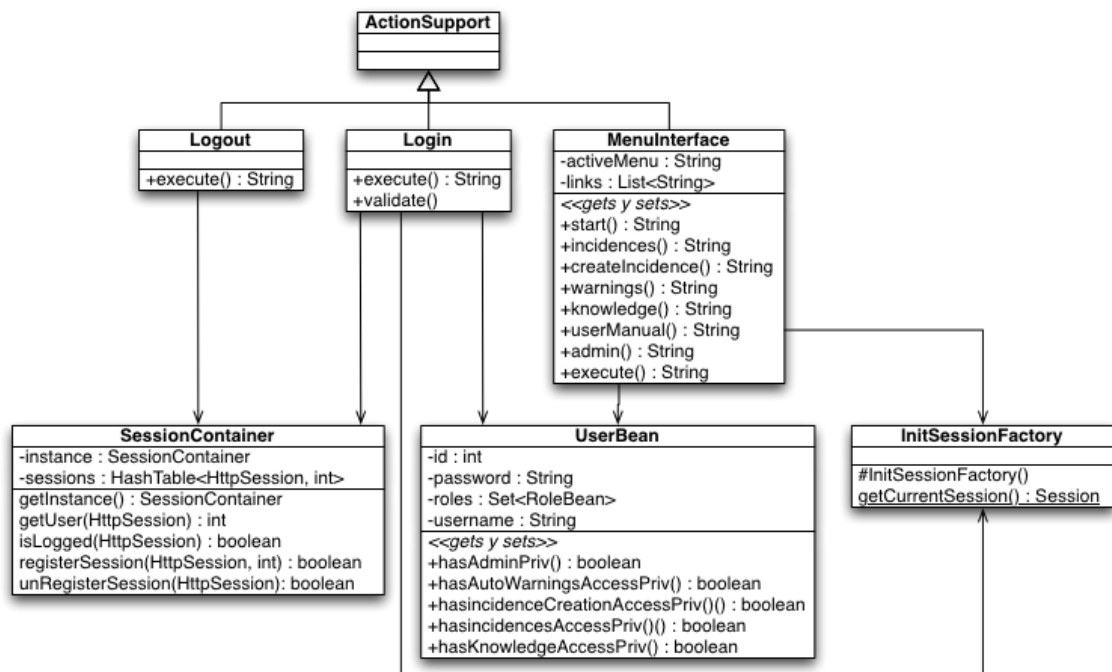


Figura 4-5:Diagrama de clases, módulo principal y de autenticación

Módulo de incidencias

El módulo de incidencias presenta el siguiente diagrama de clases:

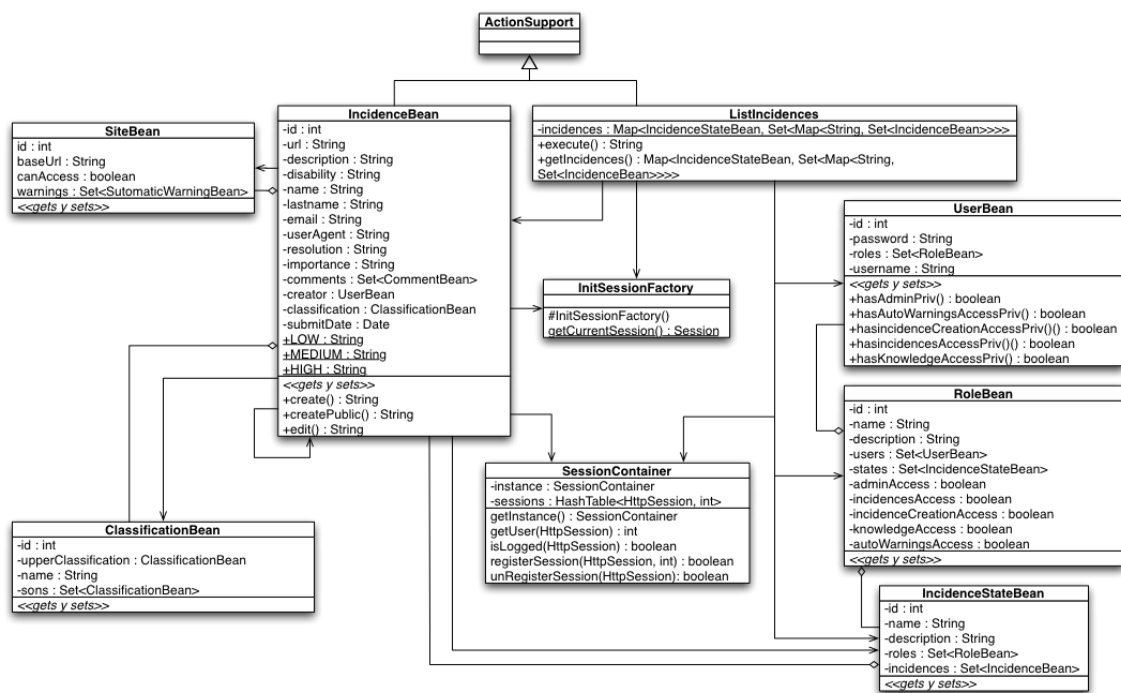


Figura 4-6:Diagrama de clases, módulo de incidencias

Módulo de alertas automáticas

El módulo de alertas automáticas presenta el siguiente diagrama de clases:

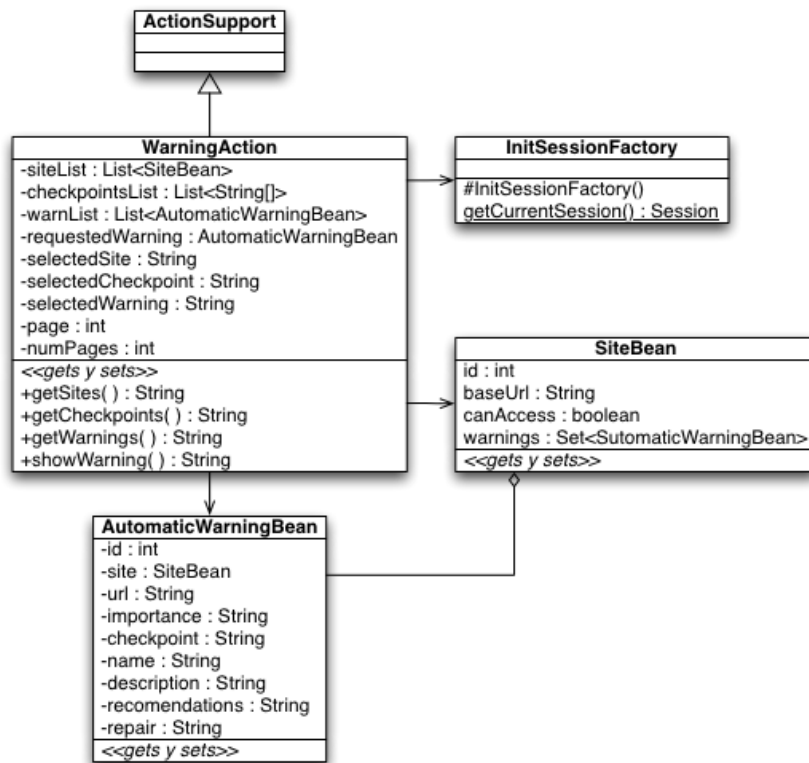


Figura 4-7:Diagrama de clases, módulo de alertas automáticas

Módulo de conocimiento

El módulo de conocimiento presenta el siguiente diagrama de clases:

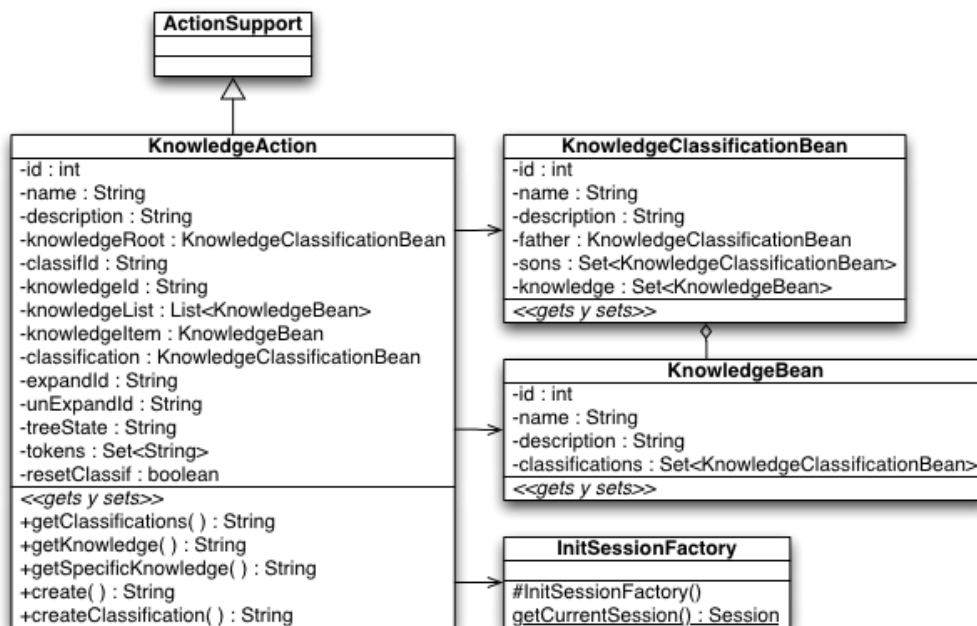


Figura 4-8:Diagrama de clases, módulo de conocimiento

Módulo de administración

El módulo de administración presenta el siguiente diagrama de clases:

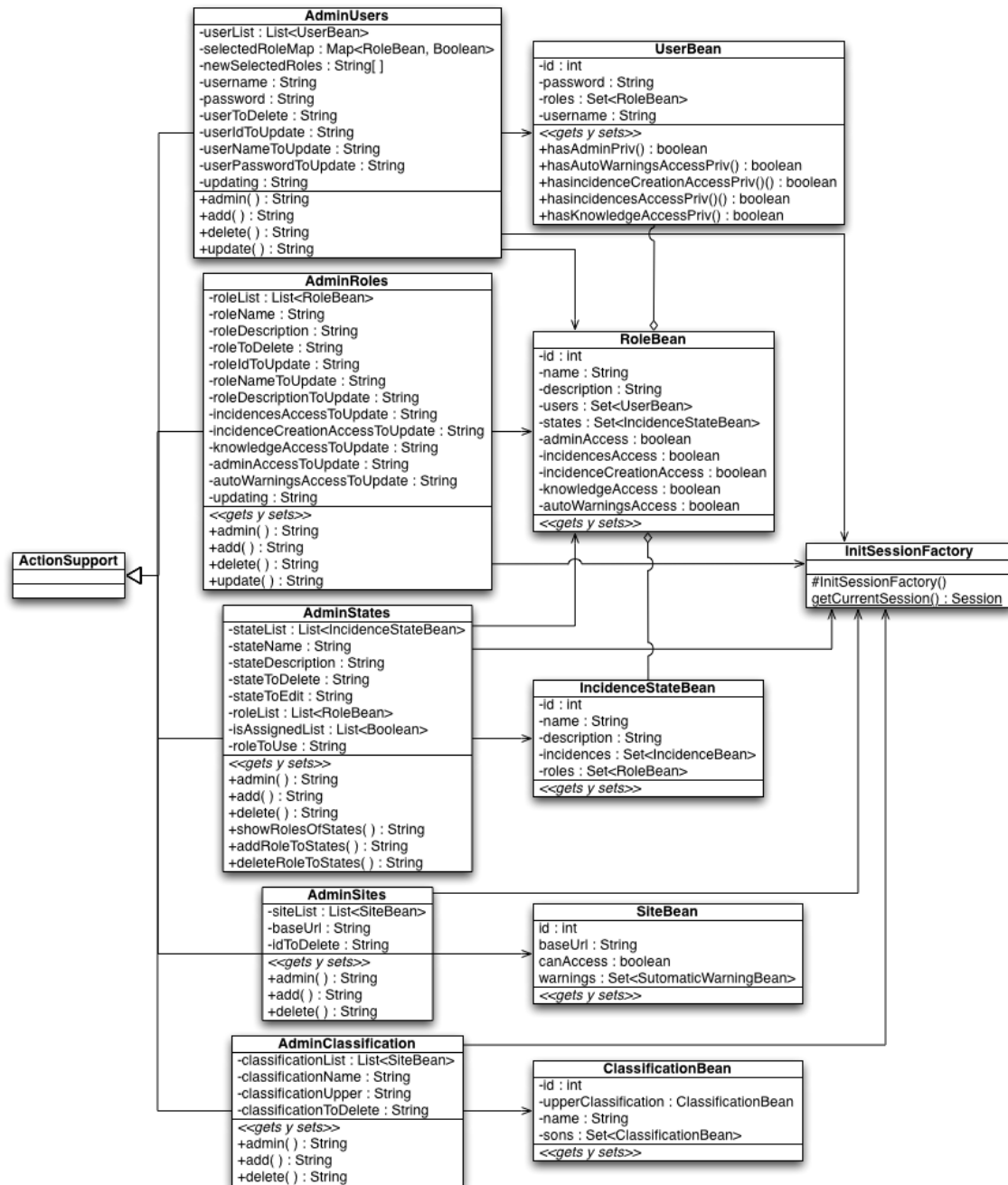


Figura 4-9: Diagrama de clases, módulo de administración

4.4.3. Mapeo Struts

Para que el *framework* de Struts sea capaz de identificar las llamadas que debe mapear, así como las clases y métodos que debe invocar para cada una de ellas y la respuesta que se debe proporcionar en función de los resultados, se debe crear un fichero "struts.xml", donde se especificarán todos estos parámetros. A continuación se detalla el proceso que sigue Struts

para el tratamiento de las llamadas, se explica la estructura del documento “struts.xml” y se proporciona un diagrama de los mapeos realizados para esta aplicación:

Funcionamiento del framework Struts2

El diagrama de la Figura 4-10 ilustra el manejo interno de las llamadas por parte del *framework* de Struts2, realizado en base a las configuración del los ficheros XML (identificados en la figura). Los pasos a seguir para cada una de las peticiones del agente de usuario seguirán la secuencia definida por los números.

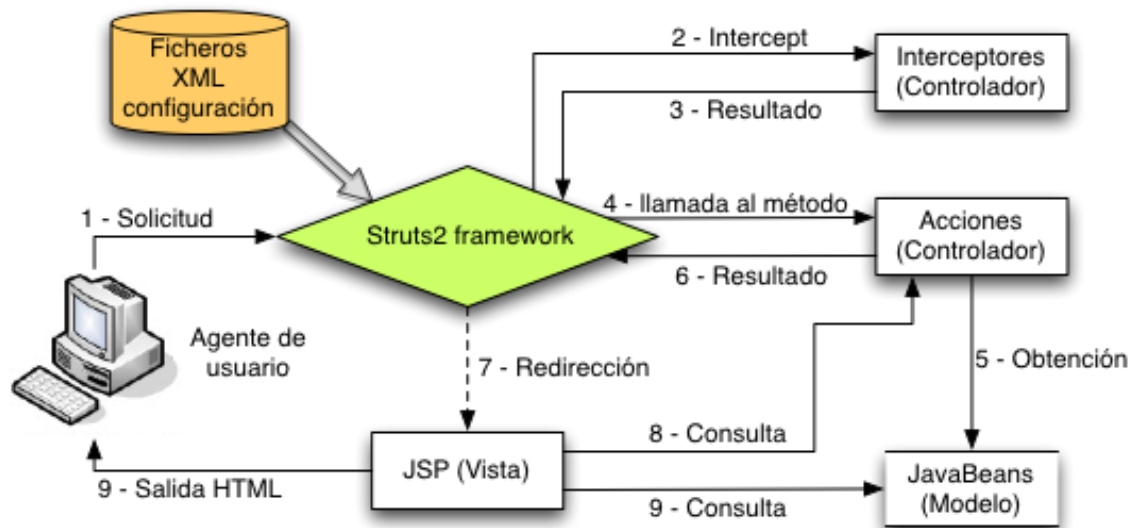


Figura 4-10: Diagrama de funcionamiento del *framework* Struts2

Proceso de tratamiento de Llamadas a La aplicación

El diagrama siguiente detalla el proceso seguido desde la invocación de una llamada de Struts hasta la obtención de la salida específica para esta llamada en forma de documento XHTML:

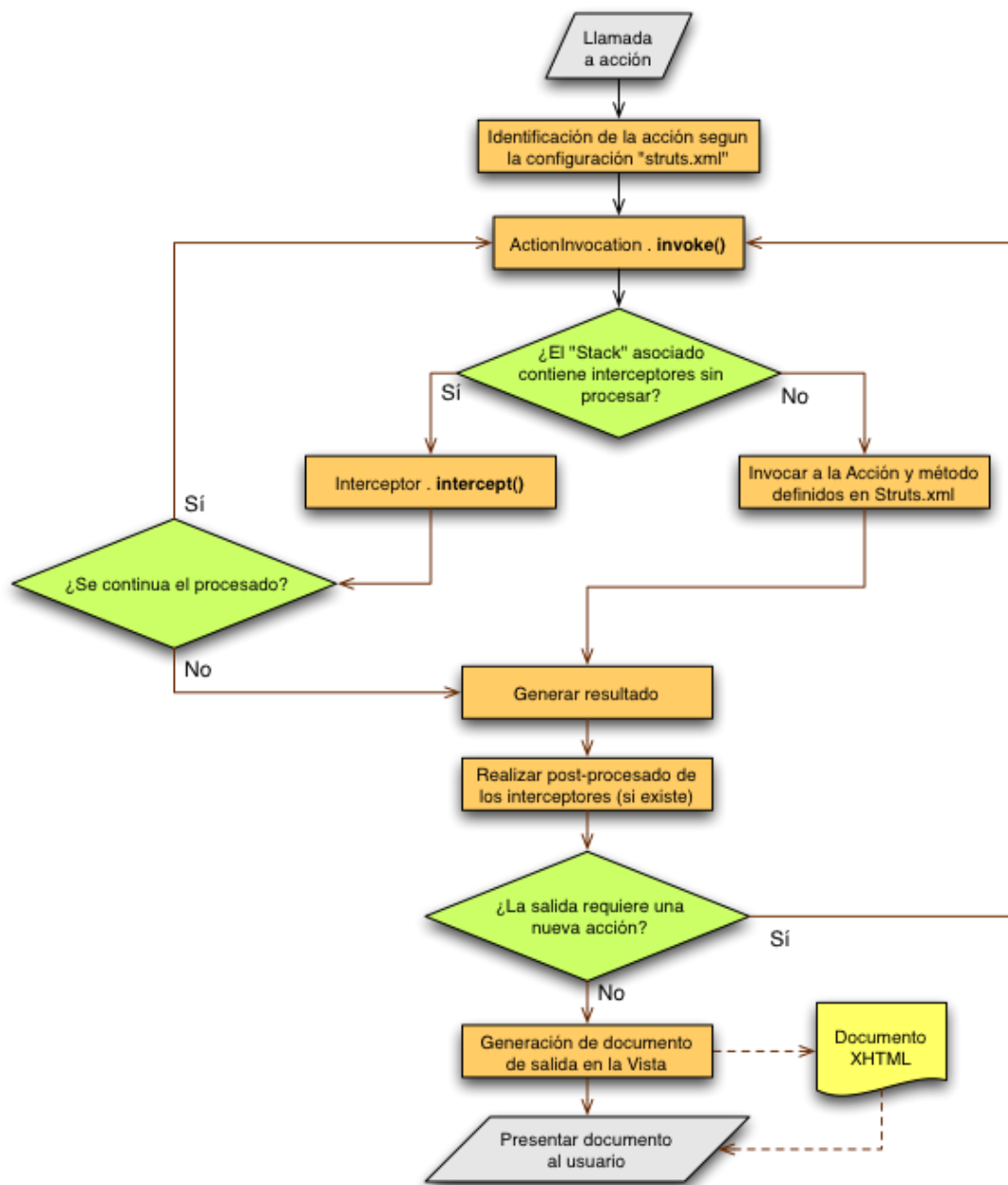


Figura 4-11: Diagrama de negocio para el procesamiento de una acción por Struts 2

Como se puede apreciar en la Figura 4-11, la primera acción llevada a cabo es la invocación del método `invoke()` de la clase "ActionInvocation", perteneciente a la librería Xwork (OpenSymphony, 2009), que es usada por Struts 2. Esta clase determina si la acción invocada tiene asociados interceptores que aun no se han procesado, y en ese caso llama al método `intercept()` del siguiente interceptor.

Si la salida del interceptor es satisfactoria, se invocará de nuevo el método `invoke()` de "ActionInvocation", continuando el proceso; en caso contrario se devolverá un valor de retorno de tipo `String`, el cual definirá la salida que se debe presentar al usuario según el componente Vista.

Si se han invocado todos los interceptores sin que ninguno de ellos detuviera la ejecución del proceso el método `invoke()` realizará la llamada a la clase y método definidos en el fichero

“struts.xml” para esta acción, que llevará a cabo los procesos necesarios y devolverá valor de retorno de tipo String, que será utilizada para identificar la salida que se debe dar en el componente Vista al igual que en el caso de los interceptores.

A continuación se realizará el post-procesado definido en los interceptores (si existe alguno) en sentido inverso a su invocación inicial, este post-procesado comprende el código situado en los interceptores después de la invocación del método `invoke()` y hereda este comportamiento de las llamadas recursivas realizadas.

Por último se identifica la operación que se debe hacer en función del resultado de la acción o del interceptor que detuvo la operación, se puede dar el caso de que el resultado requiera de la invocación de una nueva acción, lo que requerirá iniciar de nuevo el proceso; o que el resultado indique se que debe utilizar un componente de la vista para obtener la salida del usuario, en cuyo caso se generará esta en base del JSP especificado, obteniendo una salida XHTML.

Estructura del fichero “struts.xml”

A continuación se define la estructura básica del fichero:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE struts PUBLIC
    "-//Apache Software Foundation//DTD Struts Configuration 2.0//EN"
    "http://struts.apache.org/dtds/struts-2.0.dtd">

<struts>
  <package name="basicstruts2" extends="struts-default">

    <interceptors>
      <interceptor name="helloInterceptor" class="com.jar.HelloInterceptor" />

      <interceptor-stack name="helloStack">
        <interceptor-ref name="helloInterceptor" />
        <interceptor-ref name="defaultStack" />
      </interceptor-stack>
    </interceptors>

    <action name="index">
      <result>/index.jsp</result>
    </action>

    <action name="hello" class="com.jar.HelloWorldAction" method="execute">
      <result name="success">/HelloWorld.jsp</result>
    </action>

  </package>
</struts>
```

Las primeras 4 líneas definen el tipo de documento y lo identifican como un documento XML que sigue las DTD especificadas.

Dentro de los *tags* `<struts>` se ha de definir al menos un paquete (package) como se ve, el cual debe extender de “struts-default”, que es el paquete base. El resto de paquetes podrían extender tanto el paquete base como cualquier otro paquete existente.

El elemento `<interceptors>` permite definir en su interior los elementos `<interceptor>` e `<interceptor-stack>`, `<interceptor>` define un nuevo interceptor, indicando el nombre por el que se referenciará así como la clase a la que se debe invocar. `<interceptor-stack>` define nuevas pilas de interceptores, normalmente se incluyen los interceptores personalizados y después los definidos por defecto en el sistema (`defaultStack`).

En este punto se pueden definir tantos elementos `"action"` como acciones se desee implementar en el sistema, el parámetro `"name"` identifica la acción, mediante este identificador se deberá llamar a la acción en la dirección URL; el parámetro `"class"` es opcional e identifica a la clase que se invocará al ejecutar esta acción; el parámetro `"method"` también es opcional e identifica al método dentro de la clase especificada que se invocará, si no se especifica ninguno se realizará una llamada al método `"execute()"`. Estos métodos necesariamente tienen que devolver un objeto de tipo `String` y no disponer de parámetros de entrada.

Cada elemento `"action"` debe tener uno o más elementos `"result"`, los cuales identifican mediante el parámetro `"name"` el valor de retorno esperado de la clase e indican la acción a realizar en función de la respuesta. En el ejemplo anterior la acción `"index"` como resultado muestra siempre la página `"index.jsp"`, sin embargo la acción `"hello"` muestra, en caso de que su resultado (definido en la clase `"HelloWorldAction"` como un `String`) sea `"success"`, la página `"HelloWorld.jsp"`. Se pueden definir tantos resultados con un parámetro `"name"` diferente como se desee y se puede establecer uno por defecto no especificando este parámetro, el cual se seleccionará en caso de que el valor de retorno no corresponda a ninguno de los otros. En caso de que se devuelva un valor para el cual no haya un resultado definido, se generará un error.

Diagrama de estructura de la aplicación

A continuación se detalla la estructura de la aplicación mediante un diagrama que muestra las acciones del sistema, indicando sus relaciones con las clases de acción (Controlador) y con los ficheros JSP (Vista), así como las transiciones navegacionales más comunes. Debido a la naturaleza de la aplicación, este diagrama hace las veces de diseño detallado y muestra las relaciones entre todos los elementos del sistema.

Los elementos del diagrama son los siguientes:

- Elemento `"Constant"`, el cual define el nombre de los ficheros de idioma utilizados para internacionalización (i18n).
- Interceptores: Se definen los *Stack* de interceptores del sistema, en el apartado 4.4.4 Identificación de subsistemas se detalla la funcionalidad de los interceptores y en el Anexo D Detalle de los interceptores se detalla su implementación.
- Los elementos identificados como `"-Action-"` corresponden a acciones definidas en Struts que podrán o no hacer uso de clases de Acción, identificadas como `"-ActionSupport-"`. La representación de estas es la siguiente:

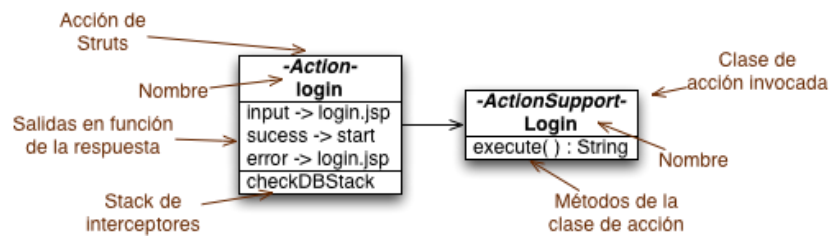


Figura 4-12: Detalle elementos Action y ActionSupport

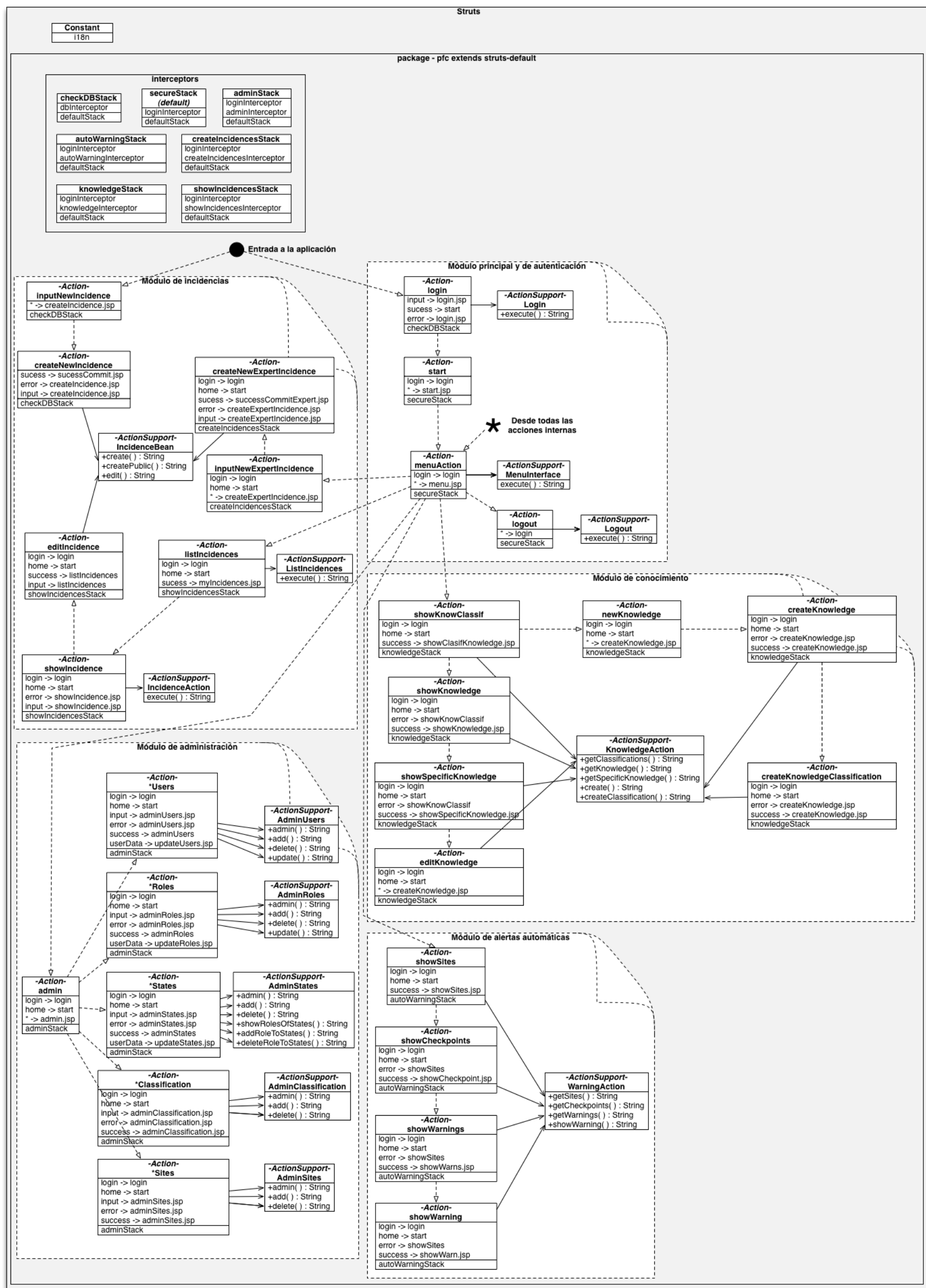


Figura 4-13: Diagrama de los mapeos de Struts para la aplicación

4.4.4. Identificación de subsistemas

Se especifica el diseño para cada uno de los subsistemas de la arquitectura especificada:

Subsistema Modelo

El subsistema modelo de la aplicación está formado por dos componentes, que son la base de datos del sistema y los JavaBeans utilizados para manejar como objetos en la aplicación la información persistente de la base de datos.

Este sistema facilita el diseño de la aplicación, agilizando el manejo de datos y asegurando la disponibilidad, además permite aprovechar las capacidades de Hibernate para recuperar información desde una base de datos relacional y presentarlos a la aplicación como objetos.

Base de datos

Para la base de datos del sistema se ha seguido como metodología de diseño el partir de un diseño relacional conceptual, y a partir de él hacer la implementación para el gestor de base de datos MySQL. En la Figura 5-14 se recoge el diagrama de este diseño.

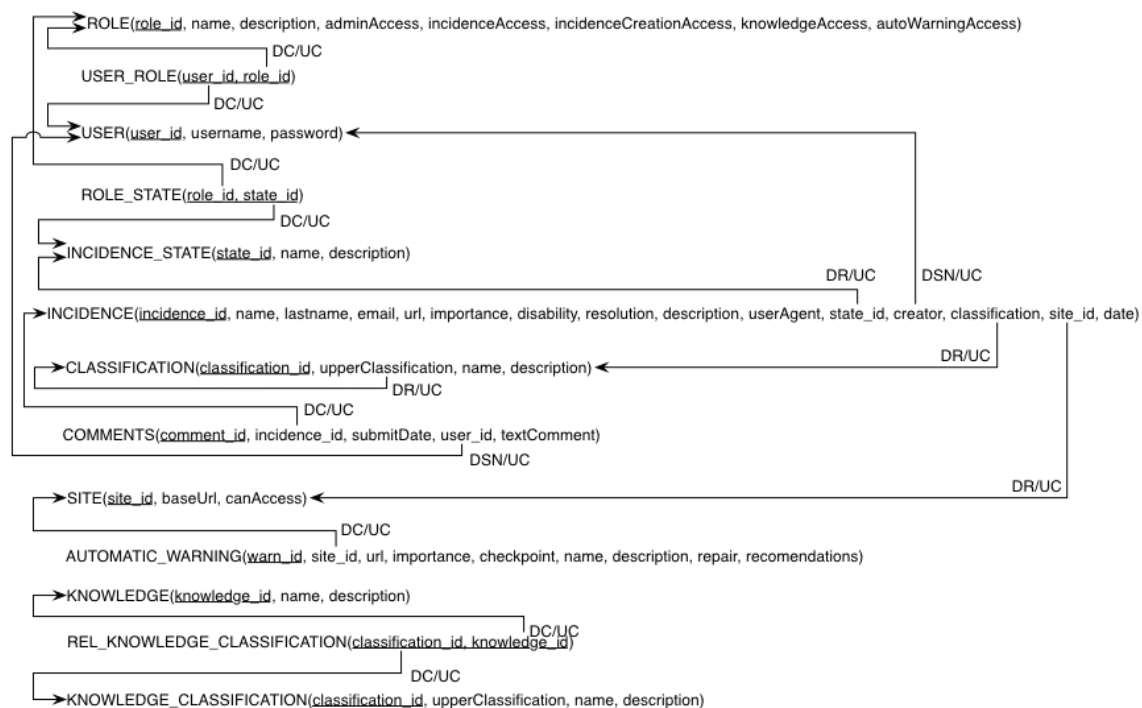


Figura 4-14: Diagrama relacional de la base de datos

Mediante este diseño de la base de datos se recogen, tanto los datos de las incidencias, ya sean automáticas o de usuario, así como su dependencia de un sitio web monitorizado y, en el caso de las incidencias de usuario, sus comentarios, estado actual, clasificación y fecha de creación. Este diseño también engloba el sistema de gestión de usuarios, almacenando roles, que podrán tener un conjunto de privilegios y usuarios, los cuales tendrán un conjunto de roles. Estas relaciones están modeladas mediante tablas intermedias, las cuales corresponden a relaciones del tipo N:M, proporcionando así la máxima flexibilidad en la configuración de cada usuario o conjunto de estos. Cabe destacar la relación entre "Role" e "Incidence_state", la cual definirá las incidencias a las que tendrá acceso cada usuario a

través de su rol. El conocimiento estará almacenado en la entidad "Knowledge" que tendrá una relación con clasificaciones similares a las incidencias, con la diferencia en este caso de que cada artículo de conocimiento puede pertenecer a varias clasificaciones, como se ha decidido en el diseño. Por último las clasificaciones estarán relacionadas con otras clasificaciones componiendo así una estructura arbórea. Asimismo se han definido las restricciones de borrado y modificación para cada relación optimizando el comportamiento de la base de datos a las necesidades de este problema particular.

JavaBeans

Como se ha expuesto, se ha hecho uso de un conjunto de JavaBeans, los cuales son usados por Hibernate para almacenar la información de la base de datos en el lado de la aplicación. A continuación se detalla la arquitectura de Hibernate y se explican los detalles:

La arquitectura de Hibernate vista desde un nivel muy abstracto, muestra el siguiente aspecto:

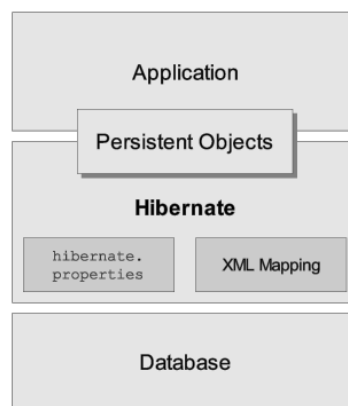


Figura 4-15: Arquitectura Hibernate de alto nivel (Red Hat, Inc, 2010)

La aplicación cuenta con unos objetos persistentes (Persistent Objects), que en este caso corresponden con los JavaBeans indicados anteriormente, los cuales son accedidos desde el *framework* de Hibernate en función de las configuraciones para cada uno de estos mapeos definidos en ficheros XML individuales (XML Mapping). Asimismo se debe proporcionar a hibernate un fichero "hibernate.cfg.xml", en el cual se establece la configuración general, como la base de datos (Database) a utilizar, el tipo de contexto de cada sesión con la base de datos y los ficheros de configuración de *mapping* de las tablas de la base de datos a objetos Java.

Una vez configurado correctamente la aplicación se comunicará con la base de datos a través de Hibernate, abstrayéndose completamente tanto de la configuración de esta como de la sintaxis para las peticiones.

Es importante destacar que Hibernate hace uso para la comunicación con la aplicación de un lenguaje basado en SQL, HQL (Red Hat, Inc, 2010). El cual presenta gracias a Hibernate total independencia de la base de datos utilizada.

Un nivel más detallado de la arquitectura de Hibernate muestra lo siguiente:

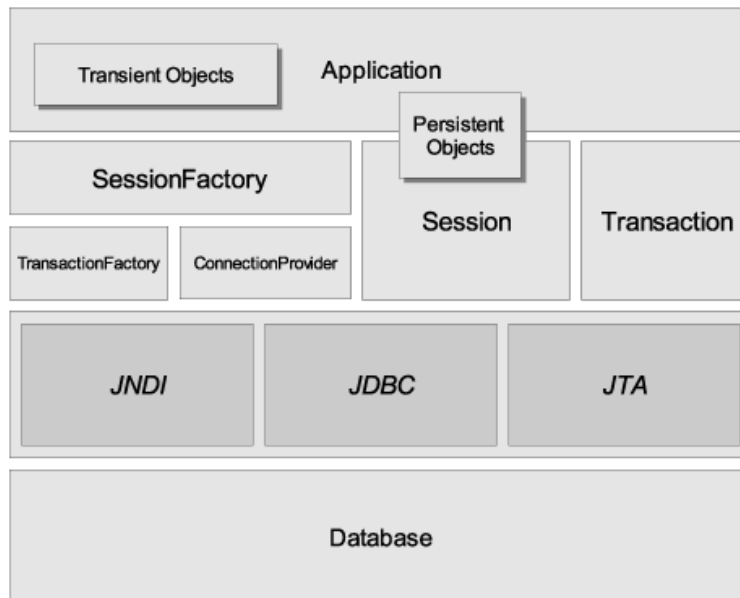


Figura 4-16: Arquitectura Hibernate detallada (Red Hat, Inc, 2010)

En este diagrama se observan objetos transitorios (Transient Objects), que no son más que objetos de la aplicación que son independientes de la base de datos y por lo tanto de Hibernate.

Es conveniente analizar el mayor detalle del *framework*:

- "SessionFactory" es un objeto *threadSafe* encargado de proporcionar a la aplicación instancias de la clase `Session` de cara a realizar operaciones contra la base de datos.
- Tanto "TransactionFactory" como "ConnectionProvider" son clases que no son accesibles desde la aplicación, sino usadas por "SessionFactory", sin embargo se podría extender o modificar su funcionamiento, trabajo que no ha sido necesario realizar.
- El objeto de la clase "Session" proporcionado por "SessionFactory" es capaz de realizar cualquier operación proporcionada por Hibernate, ya sea consultas, modificaciones o borrados de la base de datos. Esta clase representa una "conversación" con la base de datos y presenta una caché de primer nivel para las transacciones. En esta aplicación, de cara a optimizar el acceso a la base de datos, cada petición al servidor que requiera recuperar información obtiene un y solo un objeto "Session", según las recomendaciones de implementación para aplicaciones cliente-servidor.
- Cada una de las operaciones realizadas es encapsulada por Hibernate en un objeto "Transaction", abstrayendo así a la aplicación del acceso a la base de datos específico utilizado
- "JNDI", "JDBC" y "JTA" son interfaces o APIs de acceso a la base de datos, Hibernate hace uso de ellos para comunicarse con esta y están definidos en la configuración principal, sin embargo la aplicación desconoce completamente estos detalles, siendo transparentes para ella.

Una vez explicada la arquitectura del sistema se detallan los JavaBeans u objetos persistentes utilizados en la aplicación, indicando los atributos persistentes e ignorando los que no lo son, para cada uno de estos atributos se presenta un método `get` y otro `set` que siguen la nomenclatura determinada por la siguiente fórmula: `[get|set|is]Attr`, siendo `Attr` el nombre del atributo con la primera letra mayúscula.

En el Anexo A Detalle de las clases JavaBean se detalla la implementación de estas clases.

Subsistema Vista

En este sistema basado en Struts, la vista está formada por los ficheros Java Servlet Pages (JSP). Estos ficheros definen la salida XHTML del sistema, tanto para la estructura estática de esta como para el contenido dinámico, el cual es obtenido del modelo a través del controlador. Estos JSP hacen uso de las propias directivas de JSP junto con los *tags* de Struts para adecuar la información recuperada de la base de datos a la visualización necesaria en esa página web.

Como se puede apreciar en la Figura 4-13, cada acción de Struts dispondrá usualmente de varias vistas en distintos ficheros JSP, que se seleccionarán en función del resultado devuelto por los interceptores o por la clase de acción.

El cometido principal de los JSP es mostrar los datos proporcionados por la aplicación obteniéndolos desde la clase de acción a través de sus métodos `get` y enviar datos al sistema introducidos por el usuario mediante un formulario haciendo uso de los métodos `set`.

En el Anexo B Detalle de los JSP se detallan los elementos JSP de la vista de la aplicación.

Subsistema Controlador

El subsistema controlador será el encargado de manejar las peticiones de los usuarios, las cuales llegarán al sistema mediante `HttpServletRequest`. Struts es el encargado de, en base al fichero XML de configuración invocar la clase de acción apropiada en función de la entrada particular recibida.

El subsistema controlador se puede subdividir en tres partes, por un lado las clases que realizan las acciones invocadas por el usuario, por otro lado, los "Interceptores", que son clases utilizadas por Struts para realizar acciones o comprobaciones previas a la llegada de la invocación a la clase objetivo y por último un conjunto de clases Java llamadas clases auxiliares e invocadas tanto por las acciones como por los interceptores para realizar tareas complejas.

Clases de acción

Estas clases son las encargadas de realizar las acciones del sistema. Cada una de las clases representa un componente del sistema y cada uno de los métodos de este las acciones que puede realizar, siguiendo el paradigma de orientación a objetos. Struts hace uso de las librerías de Reflection, permitiendo así el acceso a los métodos definidos en estas clases aun sin conocer su definición en tiempo de compilación, esto hace de Struts una herramienta de gran potencia al desarrollar este tipo de aplicaciones.

Las clases de acción y sus objetivos son los siguientes:

- Login: Comprueba las credenciales del usuario y almacena los datos de sesión de este.
- Logout: Elimina los datos de sesión del usuario y lo saca del sistema.
- MenuInterface: Genera un menú dinámico en función de los permisos del usuario y la web en la que se encuentre.
- IncidenceBean: Proporciona métodos para crear las incidencias del sistema, tanto públicas como por expertos.
- ListIncidentes: Obtiene un listado de las incidencias del sistema.
- IncidenceAction: Proporciona métodos para modificar las incidencias del sistema.
- WarningAction: Proporciona los métodos para navegar y gestionar el módulo de incidencias automáticas.
- KnowledgeAction: Proporciona los métodos para navegar y realizar las acciones del módulo de gestión del conocimiento.
- AdminUsers: Proporciona los métodos para listar, crear, modificar y eliminar los usuarios del sistema.
- AdminRoles: Proporciona los métodos para listar, crear, modificar y eliminar los roles del sistema.
- AdminStates: Proporciona los métodos para listar, crear, modificar, eliminar y asignar roles a los estados posibles de las incidencias del sistema.
- AdminClassification: Proporciona los métodos para listar, crear y eliminar las clasificaciones posibles de las incidencias del sistema.
- AdminSites: Proporciona los métodos para listar, crear y eliminar los sitios web monitorizados por el sistema.

En el Anexo C Detalle de las clases de acción se detallan las clases Java que definen la respuesta a las llamadas desde Struts.

Interceptores

Los interceptores son clases simples que validan la petición del cliente antes de ser enviadas a la clase de acción objetivo. Su objetivo es, por ejemplo, el de comprobar que los datos de acceso introducidos en un formulario "Login" son correctos y corresponden a un usuario válido antes de llamar a la clase que autentica al usuario en el sistema.

En Struts2, los interceptores se apilan en los llamados "Stacks", en ellos se apilan un conjunto de interceptores, ejecutándose estos secuencialmente. En el caso de que alguno de ellos devuelva un error, la ejecución se detiene y se devuelve una respuesta al usuario mediante la vista, si no se produce ningún error en ninguno de ellos finalmente se ejecutará la acción objetivo. Normalmente, y en este caso no se hace excepción, los interceptores creados por el desarrollador se incluyen en el "Stack" justo después del "Stack" por defecto que proporciona Struts, el cual proporciona interceptores por defecto que proporcionan funcionalidades comunes para la mayoría de las aplicaciones.

La Figura 4-17 indica las clases que componen los interceptores del sistema, todas estas clases extienden de AbstractInterceptor.

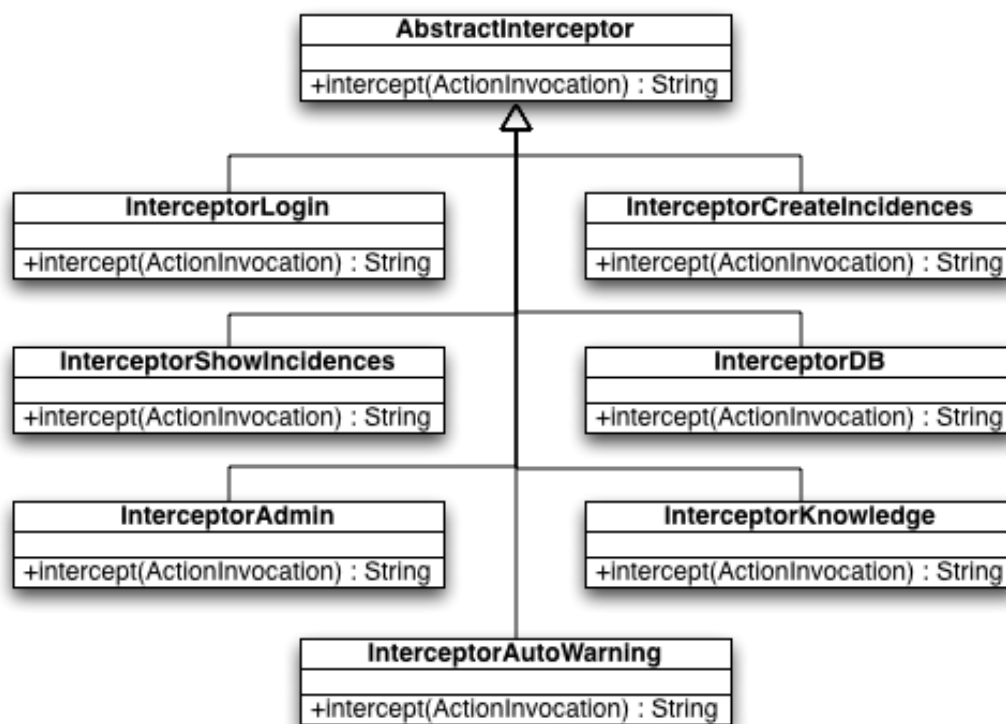


Figura 4-17:Diagrama de clases, interceptores

En el Anexo D Detalle de los interceptores, se detallan las clases Java que definen los interceptores.

Para facilitar el uso de estos interceptores en las acciones de Struts se han agrupado dentro de "Stacks", esta agrupación se detalla en el apartado Estructura del fichero "struts.xml". Cabe destacar que los identificadores utilizados en el fichero "struts.xml" para nombrar cada uno de los interceptores difiere de los nombres de las clases que implementan estos, ya que los nombres de estas clases comienzan todos por la palabra "Interceptor" de forma que fueran fácilmente reconocibles, mientras que en dicho fichero esta palabra se encuentra al final del identificador. Como ejemplo, el interceptor "loginInterceptor" corresponde a la clase "InterceptorLogin".

Clases auxiliares

Las clases auxiliares son un conjunto de elementos que implementan funcionalidad necesaria desde múltiples partes del código.

En el Anexo E Detalle de clases auxiliares se detallan las clases Java auxiliares utilizadas en la aplicación.

4.4.5. Analizador de accesibilidad (Accessibility analyzer)

El módulo del analizador de accesibilidad es el encargado de recorrer los sitios web monitorizados en busca de problemas de accesibilidad de una forma transparente para el usuario. Para ello este sistema arranca al iniciarse el servidor mediante la configuración oportuna en el fichero `web.xml`, en el que se indica que deberá iniciar automáticamente el `servlet ValidatorCrawler`.

El `HttpServlet ValidatorCrawler` actúa como un almacén que inicia una instancia de la clase `SiteManager` y proporciona métodos para salvar las alertas automáticas, borrar las alertas desfasadas y notificar a todo el sistema de obtención de alertas de la modificación de los sitios web monitorizados.

La clase `SiteManager` extiende de la clase de Java `Thread`, permitiendo una ejecución asíncrona. Su objetivo es comprobar la lista de sitios del sistema y lanzar una ejecución de la clase `Monitorizator` para cada uno de ellos, una vez terminada esta tarea la clase esperará hasta que sea notificada de la modificación de la lista de sitios monitorizados, en ese momento comprobará los cambios y arrancará hilos para los nuevos sitios o notificará a los hilos de los sitios eliminados que deben detener su ejecución. Este ciclo se realiza durante toda la ejecución del sistema.

La clase `Monitorizator` elimina las alertas antiguas de la base de datos y comienza a realizar una exploración desde la página principal del sitio. Una vez terminada esta exploración quedará en espera durante un tiempo definido hasta la próxima actualización. La exploración de los sitios se realiza haciendo uso de un método recursivo que sigue los siguientes pasos:

1. Comprueba en un `HashSet` que almacena los resúmenes hash de las direcciones URL de los sitios si el sitio actual ha sido ya visitado, en caso de que el sitio web no haya sido visitado se almacena el resumen de su dirección en este `HashSet` y se continua la ejecución.
2. Se comprueba que el tipo de contenido sea `text/html`, evitando introducir ruido al analizar ficheros de imagen u otros medios web que no sean necesarios.
3. En caso positivo se hace uso de la clase `HeraXmlConnector` que implementa el interfaz `ValidatorConnectorInterface` y que, mediante el uso del servicio de análisis de accesibilidad presenta los resultados, los cuales se almacenan en la base de datos.
4. Se recorre el contenido de la página web, en búsqueda de los enlaces a otro contenido mediante patrones. Estos enlaces se almacenan en una lista y se analizarán de forma recursiva mediante este mismo método.

La clase `HeraXmlConnector` requiere, por la implementación del sistema externo basado en el analizador Hera, la realización de dos llamadas a este, la primera realizada a en el método `getId()` y que obtiene el identificador para el análisis. La segunda realizada en el propio método `getWarnings(String)` que obtiene el fichero XML de salida del servicio externo y lo envía al método `processExit(String)` para que lo analice y genere los objetos `AutomaticWarningBean` con la información.

La clase `ACheckerConnector` funciona de forma similar a `HeraXmlConnector` salvo que contacta con el servicio de análisis de accesibilidad de `AChecker`, sin embargo no se usa actualmente, ya que la información proporcionada por este servicio es bastante más reducida a la proporcionada por Hera.

El diagrama de clases presentado a continuación ilustra el funcionamiento del sistema:

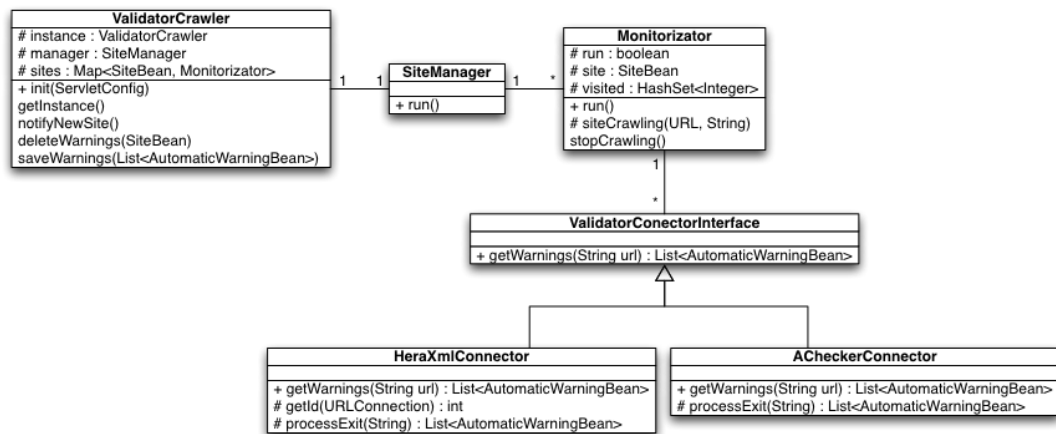




Figura 4-18: Diagrama de clases del analizador de accesibilidad

4.5. Diagramas de secuencia

A continuación se recoge un conjunto de diagramas de secuencia que ilustran la interacción de los diferentes componentes de la aplicación. Cada uno de estos diagramas muestran la actividad para una acción de usuario o un conjunto de acciones, las cuales sean realizadas normalmente juntas y en este orden, por ejemplo la obtención del formulario de autenticación y su envío.

Sobre los diagramas cabe detallar que con el objetivo de aumentar la claridad y la información aportada, se han representado tanto las clases Java como las acciones de Struts como el mismo tipo de elemento, diferenciando estos por la figura de la caja superior. Las clases Java

están representadas por una caja  y las acciones de Struts por una caja con los bordes redondeados .

4.5.1. Obtención de sesión de Hibernate

El diagrama detallado mostrado a continuación ilustra las llamadas realizadas para obtener el objeto `Session` de Hibernate. `InitSessionFactory` está implementada en el sistema como una clase no instanciable que presenta el método estático `getCurrentSession()`, el cual se encarga de realizar una llamada al método `getCurrentSession()` de la clase `SessionFactory` de Hibernate, inicializada en la creación de `InitSessionFactory`. De esta forma todas las peticiones del sistema para obtención de la una sesión pasan a través del mismo objeto `SessionFactory`, lo que evita múltiples inicializaciones del sistema y errores de Hibernate debido a la realización de múltiples inicializaciones simultaneas.

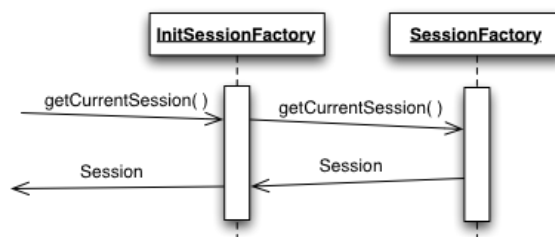


Figura 4-19: Obtención de sesión de Hibernate

4.5.2. Añadir incidencia, usuario no autenticado

El siguiente diagrama muestra el proceso seguido por el sistema cuando se introduce una incidencia desde el cliente web para usuarios no autenticados en el sistema. Se detallan a continuación los pasos seguidos:

- Se realiza una llamada a la acción `inputNewIncidence`, la cual devuelve el interfaz de la JSP `createIncidence.jsp`, que presenta el formulario de creación de incidencias para usuarios anónimos.
- Al pulsar el botón de enviar se realiza una llamada a la acción `createNewIncidence`, la cual invoca el método `create()` de la clase `IncidenceBean`, el cual obtiene la sesión de Hibernate, y gracias a esta el sitio web monitorizado que corresponde a la dirección URL aportada y el objeto correspondiente al estado de nueva incidencia. Finalmente almacena el objeto `IncidenceBean` mediante el método `Session.save()` y devuelve "success".
- Muestra el JSP `successCommit.jsp`, que indica al usuario la creación de la nueva incidencia.

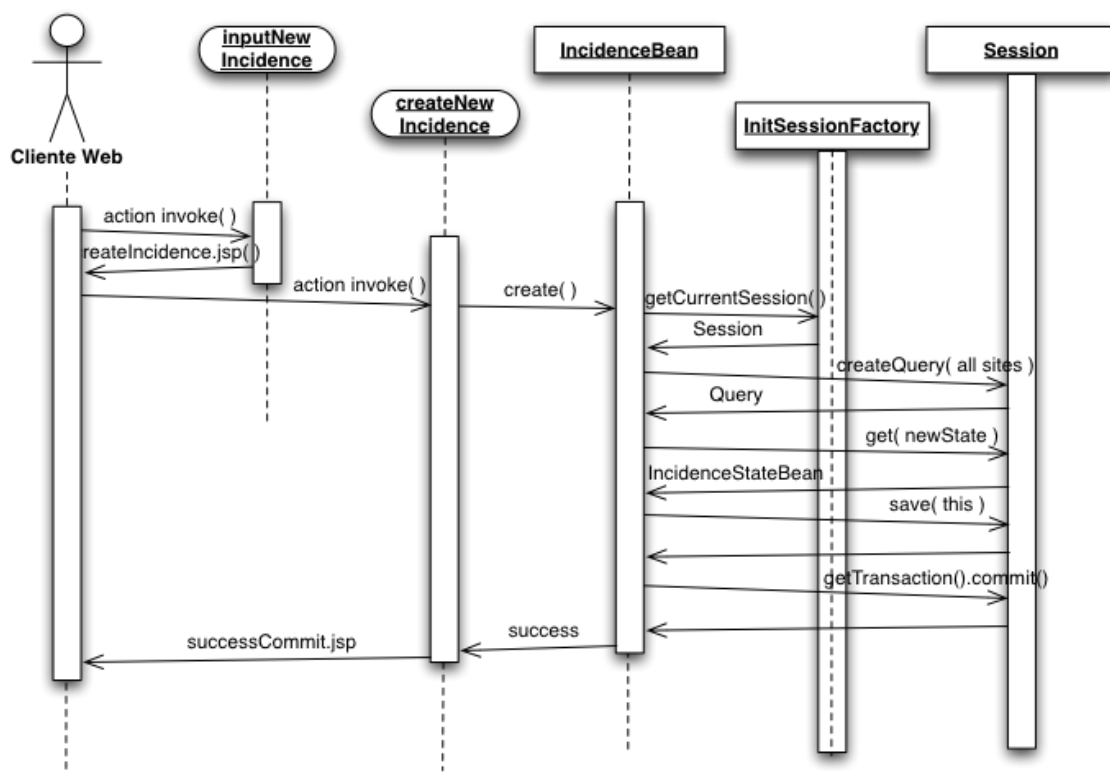


Figura 4-20: Diagrama de secuencia, añadir incidencia, usuario anónimo

4.5.3. Añadir incidencia, usuario experto

El siguiente diagrama muestra el proceso seguido por el sistema cuando se introduce una incidencia desde el cliente web por usuarios expertos, el proceso es idéntico al anterior salvo

por la obtención del usuario creador de la incidencia. Se detallas a continuación los pasos seguidos:

- Se realiza una llamada a la acción `inputNewExpertIncidence`, la cual devuelve el interfaz de la JSP `createExpertIncidence.jsp`, que presenta el formulario de creación de incidencias para usuarios anónimos.
- Al pulsar el botón de enviar se realiza una llamada a la acción `createNewExpertIncidence`, la cual invoca el método `create()` de la clase `IncidenceBean`, el cual obtiene la sesión de Hibernate, y gracias a esta el sitio web monitorizado que corresponde a la dirección URL aportada, el objeto correspondiente al estado de nueva incidencia y el objeto que representa el usuario que ha creado incidencia. Finalmente almacena el objeto `IncidenceBean` mediante el método `Session.save()` y devuelve "success".
- Muestra el JSP `successCommitExpert.jsp`, que indica al usuario la creación de la nueva incidencia.

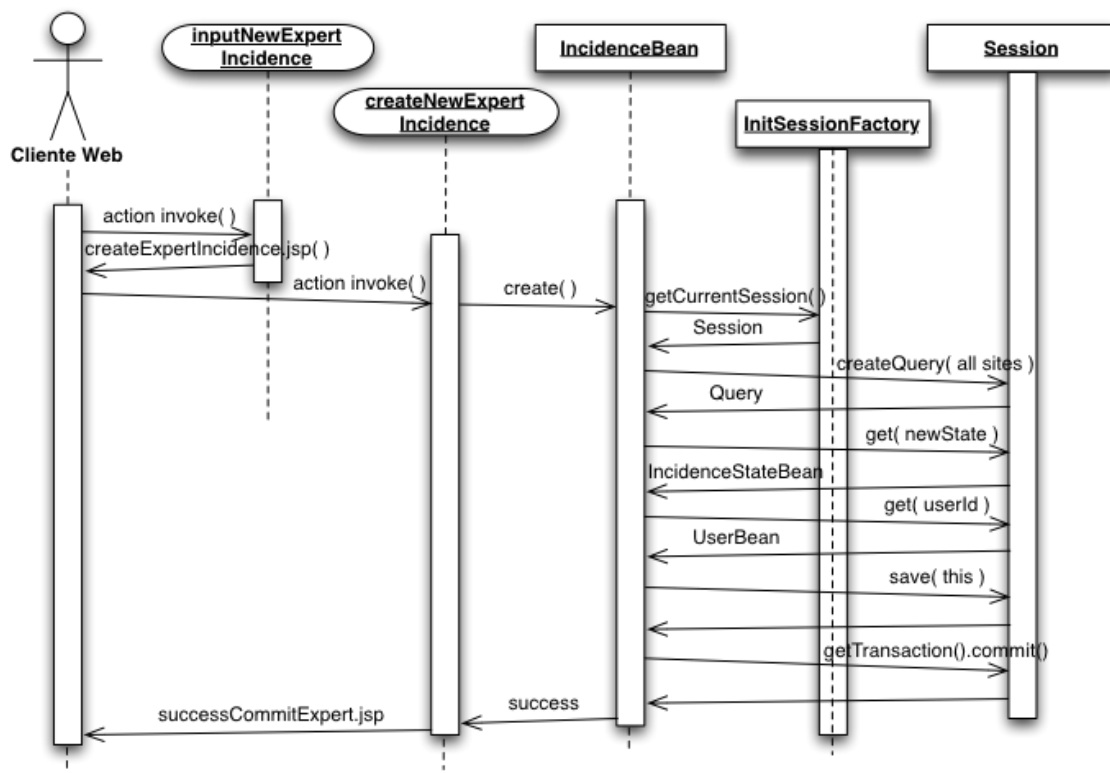


Figura 4-21: Diagrama de secuencia, añadir incidencia, usuario experto

4.5.4. Autenticarse en el sistema

El siguiente diagrama muestra el proceso seguido por el sistema cuando un usuario se autentica en este mediante el interfaz de *login*. Se detallas a continuación los pasos seguidos:

- Se realiza una llamada a la acción `login`, la cual llama al método `execute()` de la clase `Login`, ya que no hay datos de autenticación devuelve "input" y la acción a su

vez el interfaz de la JSP `login.jsp`, que presenta el formulario de autenticación de usuarios.

- Al pulsar el botón de enviar se realiza una llamada a la acción `login`, la cual invoca el método `execute()` de la clase `Login`, este devolverá "success" al contener datos de autenticación y Struts invocará al método `validate()` el cual obtiene la sesión de Hibernate, y realiza una *query* para obtener el usuario con los datos indicados, en el caso de que este exista se asocia con el objeto `HTTPSession`, permitiendo identificarlo en posteriores llamadas.
- Finalmente Struts redireccionará la salida a la acción `start`.
- Se realiza una llamada a la acción `start`, la cual devuelve el interfaz de la JSP `start.jsp`, que presenta el interfaz de bienvenida de la aplicación.

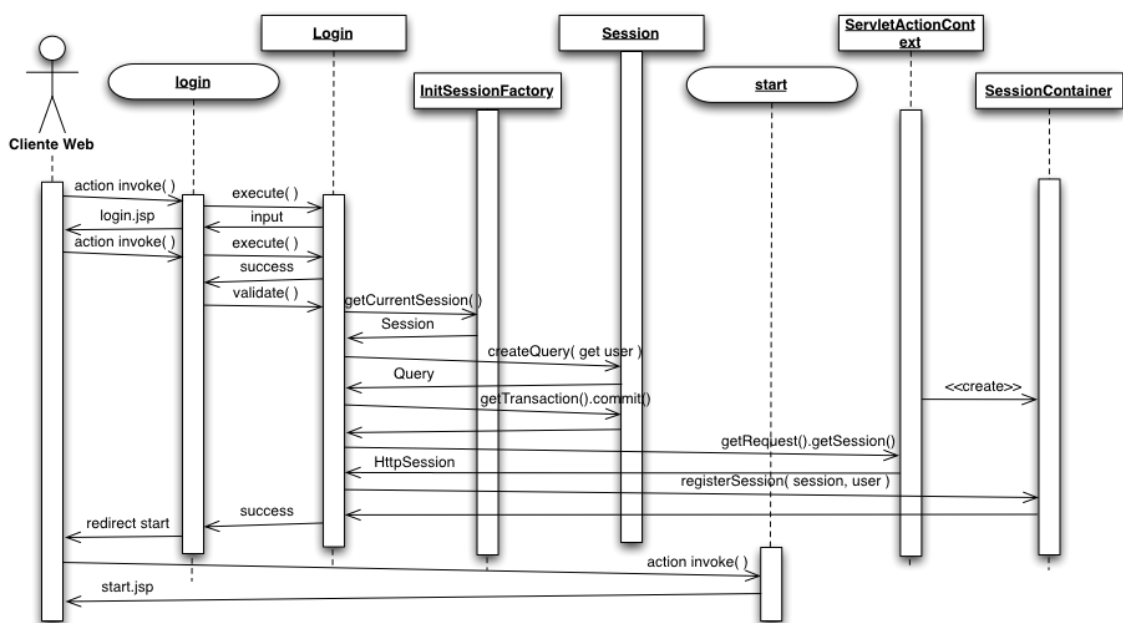


Figura 4-22: Diagrama de secuencia, autenticación de usuarios

4.5.5. Cerrar la sesión

El siguiente diagrama muestra el proceso seguido por el sistema cuando un usuario cierra su sesión mediante el menú *logout*. Se detallas a continuación los pasos seguidos:

- Se realiza una llamada a la acción `logout`, la cual llama al método `execute()` de la clase `Logout`, el cual eliminará de la lista de sesiones la asociada al la sesión actual. Finalmente Struts redireccionará la salida a la acción `login`.
- Se realiza una llamada a la acción `login`, la cual devuelve el interfaz de la JSP `login.jsp`, que presenta el interfaz autenticación de la aplicación.

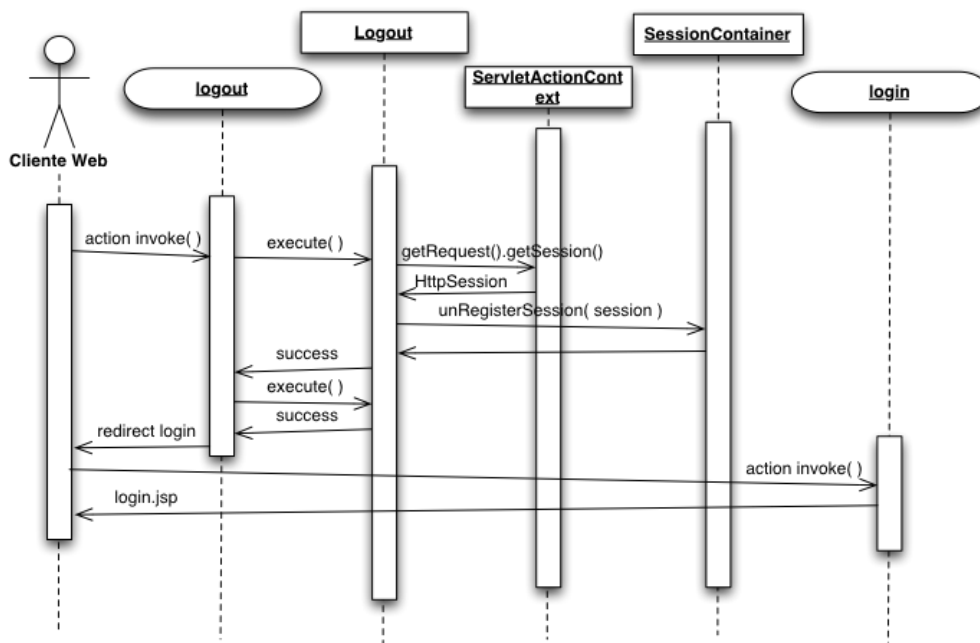


Figura 4-23: Diagrama de secuencia, autenticación de usuarios

4.5.6. Listado de incidencias

El siguiente diagrama muestra el proceso seguido por el sistema cuando un usuario lista sus incidencias. Se detallan a continuación los pasos seguidos:

- Se realiza una llamada a la acción `listIncidences`, la cual llama al método `execute()` de la clase `ListIncidences` la cual obtiene la sesión de Hibernate, y realiza una *query* para obtener el objeto `UserBean` con los datos del usuario actual, a partir de este se obtienen los objetos `RoleBean` asociados que representan sus roles y mediante ellos los objetos `StateBean`, que serían los estados de incidencias a los que el usuario tiene acceso, obteniendo en base a estos los objetos `IncidenceBean` que representan las incidencias en sí. Finalmente con toda esta información se genera una estructura que almacena el conjunto de los datos como campo de la clase y se devuelve "success".
- Struts devuelve como resultado el interfaz de la JSP `myIncidences.jsp`, que presenta el interfaz de listado de incidencias.
- `myIncidences.jsp` realizar una llamada al método `getIncidences()` del objeto de la clase `ListIncidences` anteriormente utilizado, obteniendo la estructura de objetos comentada y procesandola para presentarla en el interfaz.

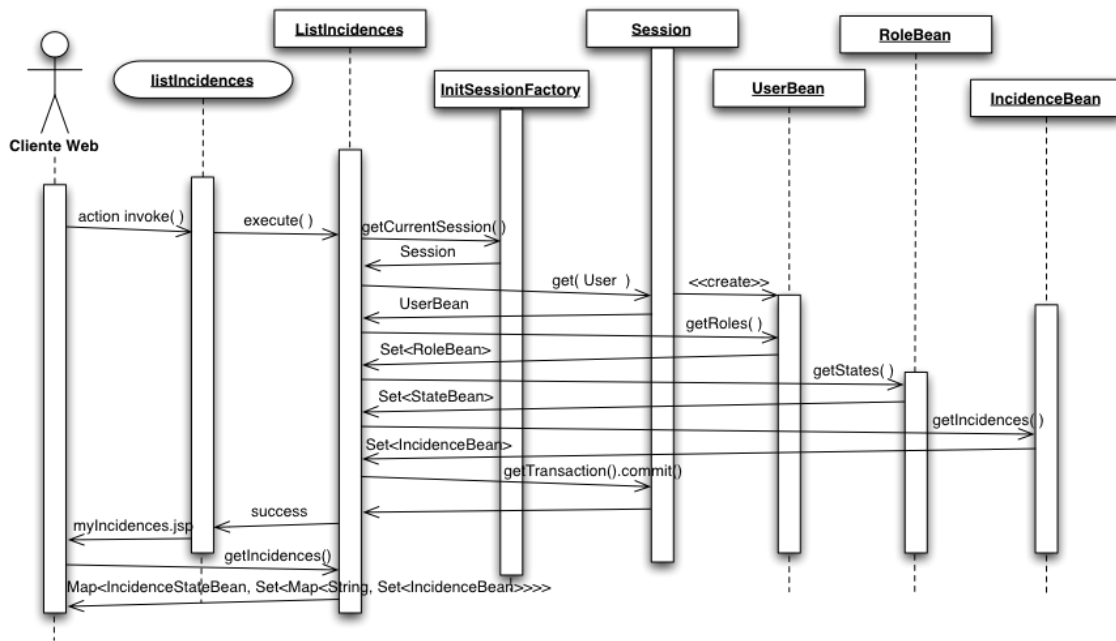


Figura 4-24: Diagrama de secuencia, listado de incidencias

4.5.7. Visualización de incidencia

El siguiente diagrama muestra el proceso seguido por el sistema cuando un visualiza una incidencia determinada. Para acceder a la visualización de incidencias se realiza anteriormente la acción de listado de incidencia. Se detallas a continuación los pasos seguidos:

- Se realiza una llamada a la acción `showIncidence`, la cual llama al método `execute()` de la clase `IncidenceAction` la cual obtiene la sesión de Hibernate, y realiza una *query* para obtener el objeto `IncidenceBean` con los datos de la incidencia especificada, de igual forma se obtienen todas las clasificaciones del sistema y todos los estados, de forma que estos se puedan consultar desde la vista para cambiar el valor actual para la incidencia. Toda esta información se almacena en los campos de la clase y se devuelve "success".
- Struts devuelve como resultado el interfaz de la JSP `showIncidence.jsp`, que presenta el interfaz de visualización de la incidencia.
- `showIncidence.jsp` realizar una llamada a los métodos `getIncidence()`, `getClassifications()` y `getStateList()` del objeto de la clase `IncidenceAction` anteriormente utilizado, obteniendo los objetos necesarios para la presentación en el interfaz.

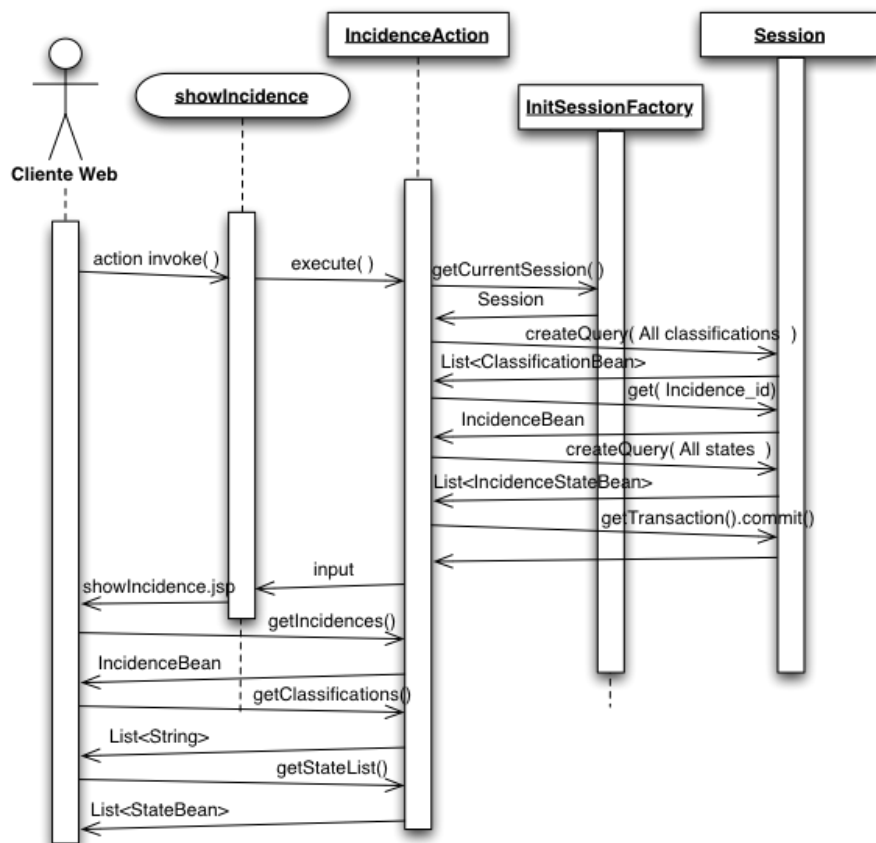


Figura 4-25: Diagrama de secuencia, visualización de incidencia

4.5.8. Edición de incidencia

El siguiente diagrama muestra el proceso seguido por el sistema cuando un visualiza una incidencia determinada. Para acceder a la edición de la incidencia se realiza anteriormente la acción de visualización de la incidencia. Se detallas a continuación los pasos seguidos:

- Se realiza una llamada a la acción `editIncidence`, la cual llama al método `edit()` de la clase `IncidenceBean` la cual obtiene la sesión de Hibernate, y realiza una *query* para obtener el objeto `IncidenceBean` asociado con el identificador de la incidencia que se desea modificar. Una vez obtenido este objeto se modifican sus atributos a través de sus métodos `set` y se actualiza la base de datos mediante `update()`, devolviendo el valor "success".
- Struts redirige la salida a la acción de listado de incidencias, `listIncidences`.

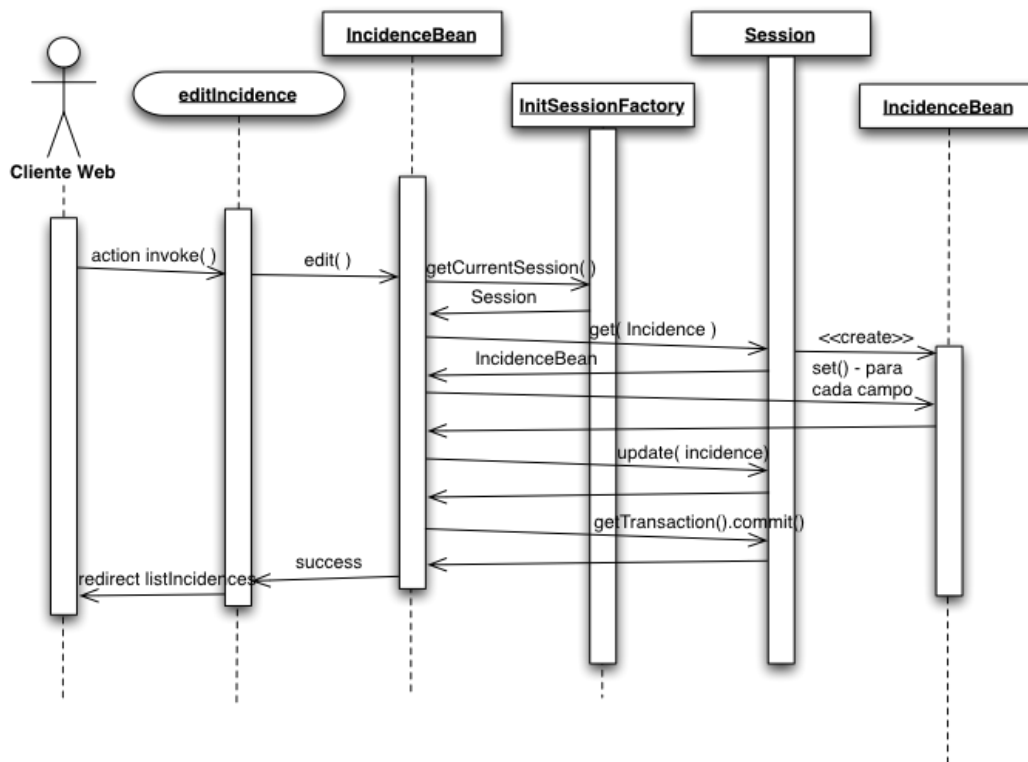


Figura 4-26: Diagrama de secuencia, edición de incidencia

4.5.9. Ver alertas automáticas, listado de sitios monitorizados

El siguiente diagrama muestra el proceso seguido por el sistema cuando un usuario lista los sitios monitorizados para visualizar las alertas automáticas. Se detallan a continuación los pasos seguidos:

- Se realiza una llamada a la acción `showSites`, la cual llama al método `getSites()` de la clase `WarningAction` la cual obtiene la sesión de Hibernate, y realiza una *query* para obtener el listado de todos los sitios monitorizados del sistema, como un `List<SiteBean>`, el cual almacenará en el campo de la clase, devolviendo posteriormente el valor "success".
- Struts devuelve como resultado el interfaz de la JSP `showSites.jsp`, que presenta el interfaz de listado de incidencias.
- `showSites.jsp` realizar una llamada al método `getSiteList()` del objeto de la clase `WarningAction` anteriormente utilizado, obteniendo la lista de objetos `SiteBean` y procesándolos para crear la salida.

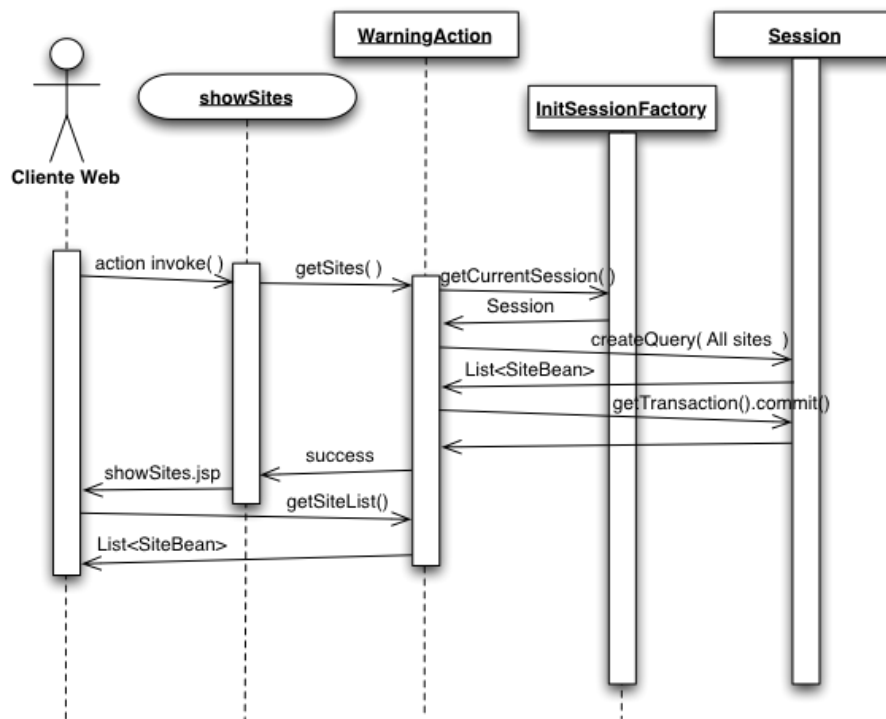


Figura 4-27: Diagrama de secuencia, ver alertas automáticas, listado de sitios monitorizados

4.5.10. Ver alertas automáticas, listado de puntos de verificación

El siguiente diagrama muestra el proceso seguido por el sistema cuando un usuario lista los puntos de verificación para el sitio monitorizado indicado. Para acceder al listado de puntos de verificación se realiza anteriormente la acción de listado de los sitios web monitorizados. Se detallan a continuación los pasos seguidos:

- Se realiza una llamada a la acción `showCheckpoints`, la cual llama al método `getCheckpoints()` de la clase `WarningAction` la cual obtiene la sesión de Hibernate, y realiza una *query* para obtener el listado de los puntos de verificación que contienen alguna alerta para el sitio monitorizado especificado, como un `List<String[]>`, el cual almacenará en el campo de la clase, devolviendo posteriormente el valor "success".
- Struts devuelve como resultado el interfaz de la JSP `showCheckpoints.jsp`, que presenta el interfaz de listado de los puntos de verificación.
- `showCheckpoints.jsp` realizar una llamada al método `getCheckpointsList()` del objeto de la clase `WarningAction` anteriormente utilizado, obteniendo la lista de los puntos de verificación y procesándolos para crear la salida.

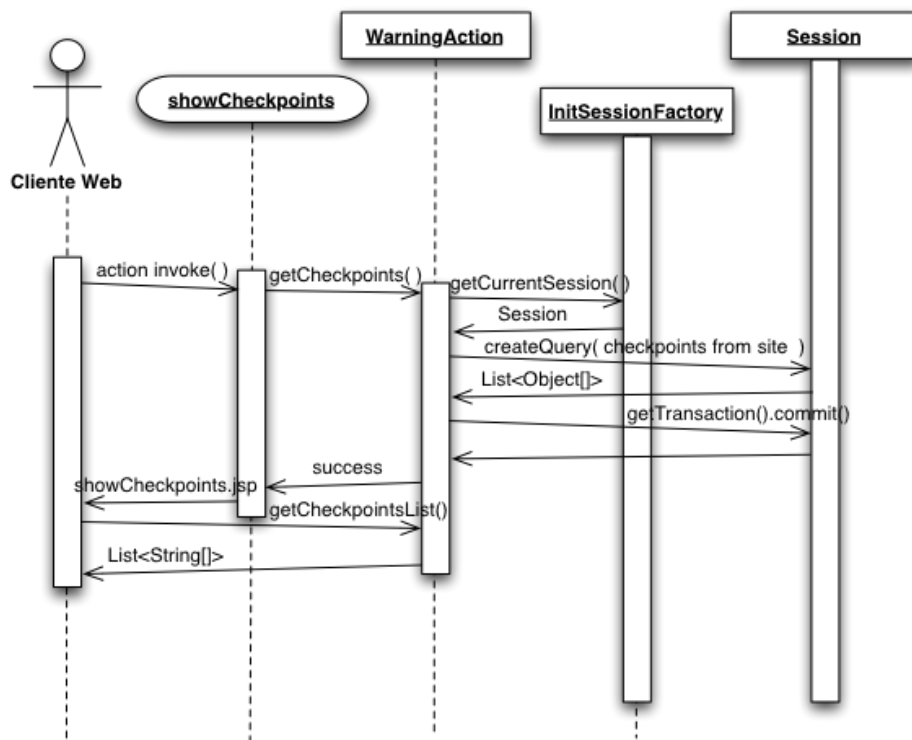


Figura 4-28: Diagrama de secuencia, ver alertas automáticas, listado de puntos de verificación

4.5.11. Ver alertas automáticas, listado de alertas de accesibilidad

El siguiente diagrama muestra el proceso seguido por el sistema cuando un usuario lista las alertas de accesibilidad para el sitio monitorizado indicado y un punto de verificación determinado. Para acceder al listado de alertas automáticas se realiza anteriormente la acción de listado de los puntos de verificación. Se detallan a continuación los pasos seguidos:

- Se realiza una llamada a la acción `showWarnings`, la cual llama al método `getWarnings()` de la clase `WarningAction` la cual obtiene la sesión de Hibernate, y realiza una *query* para obtener el listado de las alerta automáticas para el sitio monitorizado y el puntos de verificación especificados, como un `List<AutomaticWarningBean>`, el cual almacenará en el campo de la clase, devolviendo posteriormente el valor "success".
- Struts devuelve como resultado el interfaz de la JSP `showWarns.jsp`, que presenta el interfaz de listado de las alertas automáticas.
- `showWarns.jsp` realiza una llamada al método `getWarnList()` del objeto de la clase `WarningAction` anteriormente utilizado, obteniendo la lista de las alertas automáticas y procesándolas para crear la salida.

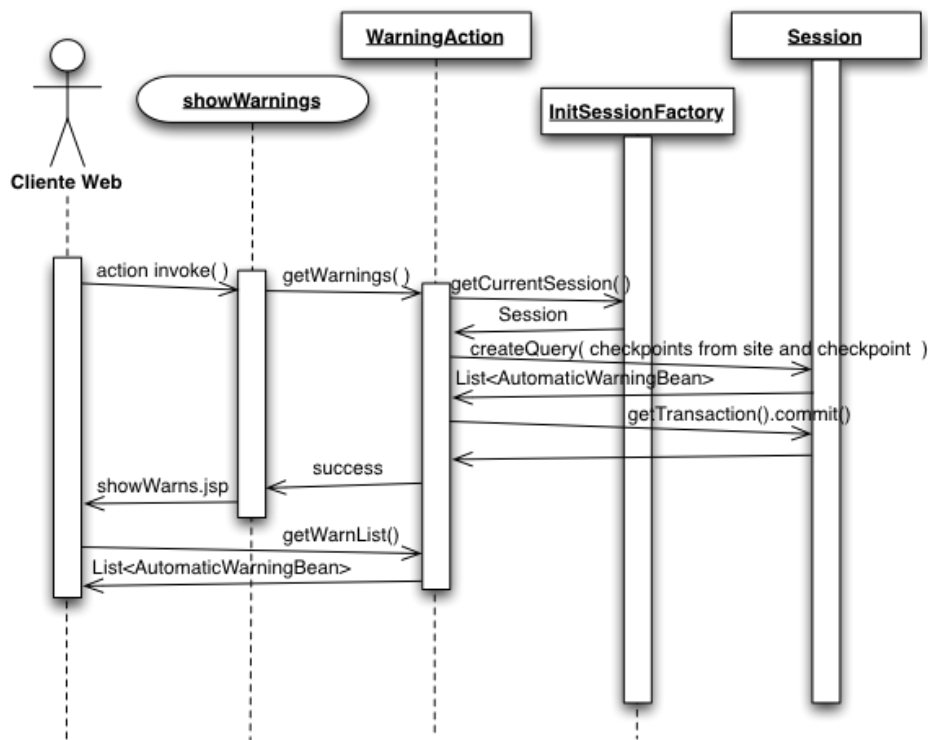


Figura 4-29: Diagrama de secuencia, ver alertas automáticas, listado de puntos de verificación

4.5.12. Ver alertas automáticas, visualizar alerta de accesibilidad

El siguiente diagrama muestra el proceso seguido por el sistema cuando un usuario visualiza la alerta de accesibilidad seleccionada. Para acceder a la visualización de incidencia se realiza anteriormente la acción de listado de las alertas. Se detallan a continuación los pasos seguidos:

- Se realiza una llamada a la acción `showWarning`, la cual llama al método `showWarning()` de la clase `WarningAction` la cual obtiene la sesión de Hibernate, y realiza una *query* para obtener el objeto `AutomaticIncidenceBean` solicitado, el cual almacenará en el campo de la clase, devolviendo posteriormente el valor "success".
- Struts devuelve como resultado el interfaz de la JSP `showWarn.jsp`, que presenta el interfaz de listado de las alertas automáticas.
- `showWarn.jsp` realiza una llamada al método `getSpecificWarn()` del objeto de la clase `WarningAction` anteriormente utilizado, obteniendo el objeto de incidencia automática y procesándolo para mostrarlo en el interfaz.

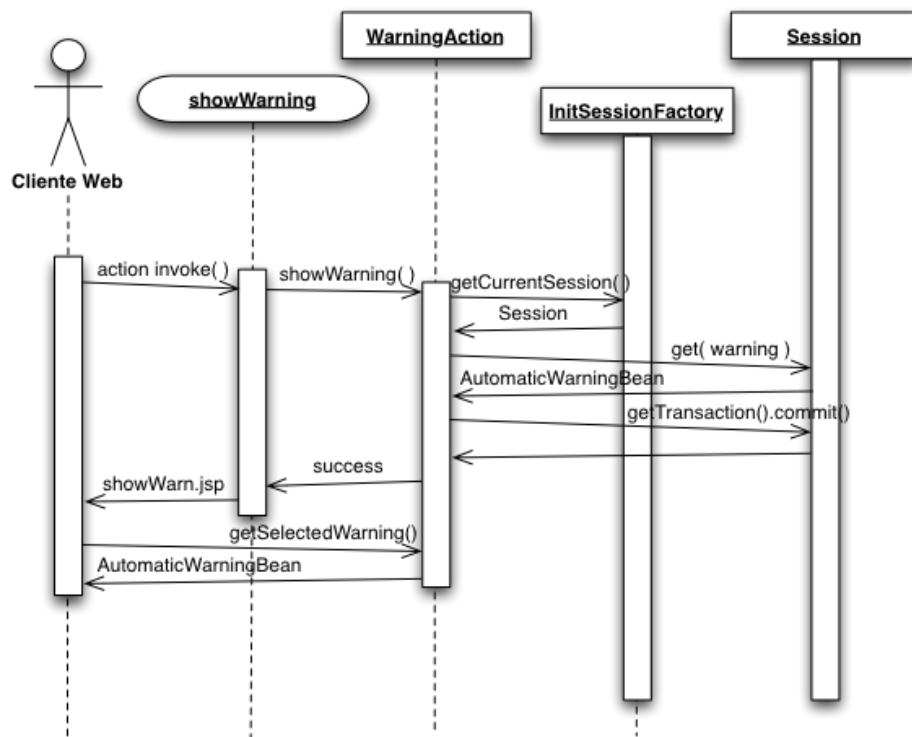


Figura 4-30: Diagrama de secuencia, ver alertas automáticas, visualizar alerta de accesibilidad

Capítulo 5.

Manual de usuario

SIGA, el Sistema Integral de Gestión de la Accesibilidad proporciona una interfaz unificada para todos los usuarios gestores del sistema, independientemente de los roles de los que estos dispongan.

A continuación se muestra todas las opciones que presenta el sistema, las opciones particulares presentadas para cada usuario dependerán de los permisos de los que disponga este.

Índice de Contenidos

5.1.	Controles comunes.....	92
5.2.	Acceso a la aplicación	93
5.3.	Pantalla de bienvenida de la aplicación	94
5.4.	Gestión de incidencias.....	94
5.5.	Creación de una nueva incidencia.....	98
5.6.	Visualización de alertas automáticas de accesibilidad	99
5.7.	Gestión de la base de conocimiento	102
5.8.	Administración del sistema	106

5.1. Controles comunes

A continuación se explican los controles más comunes que aparecen en todas las interfaces de la aplicación. Estos controles están marcados en la Figura 5-1 mediante óvalos e identificados por las letras definidas a continuación:

- A. Enlace al inicio de la aplicación mediante en enlace de cabecera.
- B. Menú principal de navegación.
- C. Interfaz principal de la aplicación, en ella se muestra la información y se introducen los datos por el usuario.
- D. Miguitas para la navegación por la aplicación.
- E. Selector de idioma.

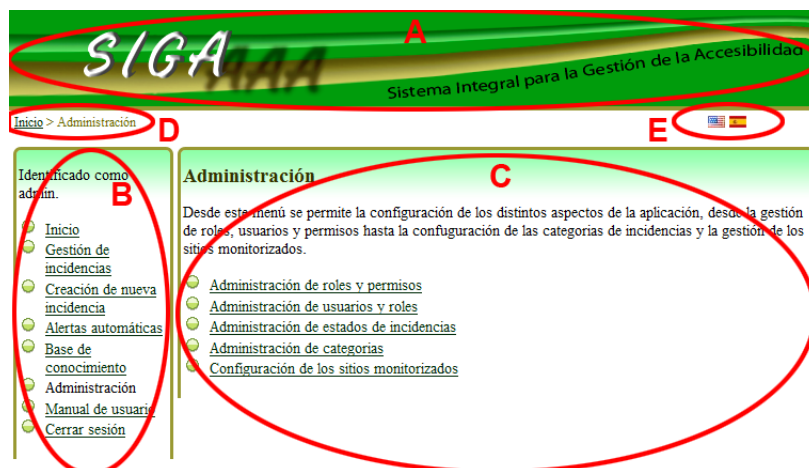


Figura 5-1: Controles comunes

5.1.1. Mensajes al usuario

Cuando se realice alguna acción que modifique el estado de la aplicación se mostrará al usuario un mensaje de confirmación o de error, el mismo formato se utiliza en los mensajes de error de autenticación. Estos mensajes se muestran a continuación:

- Mensaje de confirmación

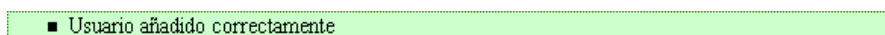


Figura 5-2: Mensaje de confirmación

- Mensaje de error



Figura 5-3: Mensaje de error

5.2. Acceso a la aplicación

La interfaz de acceso al portal de gestión de la aplicación presenta el menú de selección de idioma en la parte superior derecha, así como el formulario de acceso, en el que se deberá indicar tanto el nombre de usuario como la contraseña (marcados con óvalos) y pulsar sobre el botón enviar para acceder.



Figura 5-4: Interfaz de acceso a la aplicación

5.3. Pantalla de bienvenida de la aplicación

Al acceder al sistema o al pulsar sobre el enlace "Inicio" del menú principal de la aplicación se mostrará la pantalla de bienvenida de la aplicación (Figura 5-5) que contiene una breve introducción al sistema y permite al acceso al menú principal de la aplicación (marcado con un ovalo), el cual estará disponible durante toda la navegación del usuario en el sistema. Los elementos de este menú dependerán de los permisos de los que disponga el usuario.



Figura 5-5: Interfaz de bienvenida

5.4. Gestión de incidencias

Este módulo del sistema permite consultar y modificar las incidencias incluidas tanto por los usuarios de la aplicación como por usuarios expertos.

Al pulsar sobre el enlace "Gestión de incidencias" del menú principal de la aplicación se muestra un listado de las incidencias clasificadas según su estado y el sitio web monitorizado al que pertenecen (marcados con un ovalo) (Figura 5-6). Este listado muestra el identificador de la incidencia, su descripción, la clasificación actual de esta y la importancia mediante un icono distintivo.

SIGA
Sistema Integral para la Gestión de la Accesibilidad

Inicio > Gestión de incidencias

Identificado como admin.

- Inicio
- Gestión de incidencias
- Creación de nueva incidencia
- Alertas automáticas
- Base de conocimiento
- Administración
- Manual de usuario
- Cerrar sesión

Gestión de incidencias

Interfaz destinado a la gestión de las incidencias vinculadas a los sitios web.

A continuación se muestra tanto un resumen del número de incidencias en cada estado como la relación de incidencias, clasificadas por estado y por sitio web.

Estado new

Incidencias del estado new

Sitio: <http://www.uc3m.es>

Identificador de incidencia	Descripción	Clasificación	Importancia
2	No se puede acceder con lector de pantalla	No clasificada	
3	Problemas de formulario	general	
4	El contenido multimedia no es accesible	No clasificada	

Estado analyzing

Incidencias del estado analyzing

Sitio: <http://www.uc3m.es>

Identificador de incidencia	Descripción	Clasificación	Importancia
1	No funciona desplegable seleccionable "Seleccionar organismo" sin el java script esta desactivado	general	

© 2010 Adrián López

W3C XHTML 1.0 W3C WAI-AAA WCAG 1.0

Figura 5-6: Interfaz de gestión de incidencias

Pulsando sobre el identificador de una incidencia (marcados con óvalos) se accederá a la visualización y edición de la totalidad de sus campos, como el detalle mostrado en la Figura 5-7.

Administración

Manual de usuario

Cerrar sesión

Sitio: <http://www.uc3m.es>

Identificador de incidencia

2	No se puede
3	No se puede
6	Problemas de formulario

Figura 5-7: Detalle de acceso a visualización y edición de incidencias

5.4.1. Visualización y edición de incidencias

Desde el interfaz de gestión de incidencias se accederá al interfaz de visualización y edición pulsando sobre el identificador de la incidencia.

La interfaz de visualización y edición de incidencias (Figura 5-8) muestra un conjunto de controles con la información actual de esta, la cual podrá ser modificada por el usuario. Cabe destacar que para eliminar una incidencia se deberá marcar la casilla "¿Borrar incidencia?" en la parte inferior del formulario (marcado con un ovalo). Para enviar cualquier cambio al servidor se deberá pulsar el botón "Enviar", de igual manera se podrá pulsar el botón "Restaurar" para devolver los valores de los campos a su estado original (ambos marcados con un ovalo). En cualquier caso los campos URL y Descripción siempre deberán tener un valor no

vacío para que se acepten las modificaciones enviadas, en caso contrario se presentará un error al usuario.

SIGA
Sistema Integral para la Gestión de la Accesibilidad

Inicio > Gestión de incidencias > Detalle de incidencia

Identificado como admin.

- Inicio
- Gestión de incidencias
- Creación de nueva incidencia
- Alertas automáticas
- Base de conocimiento
- Administración
- Manual de usuario
- Cerrar sesión

Detalle de incidencia

Estado de la incidencia
Selección de estado: analyzing

Datos del problema
Identificador: 5
Url: * http://www.uc3m.es
Descripción: * El contenido multimedia no es accesible

Fecha de creación: 2010-07-06 11:18:26.0

Importancia: Media
Clasificación: general

Datos de usuario
Discapacidad: Ceguera parcial
Nombre:
Apellidos:
Email: asd@asd.com

Datos de modo de acceso
Navegador: Internet explorer
Resolución: 1024x768

Comentarios
admin 06/07/2010 - 11:27
"Revisando"
Nuevo comentario:

Edición
¿Borrar incidencia? ☐
Enviar Restaurar

*: Campos requeridos

© 2010 Adrian López

WCAG 1.0 XHTML 1.0 WAI-AAA

Figura 5-8: Interfaz de visualización y edición de incidencias

Detalles del interfaz de visualización y edición de incidencias

Se podrá modificar el estado de la incidencia a través del menú desplegable que se muestra a continuación, el cual se encuentra disponible como primer campo del interfaz, en el apartado "Estado de la incidencia", como se muestra en el detalle de la Figura 5-9.

Nota: Los elementos disponibles en este menú dependerán de los estados configurados para las incidencias y pueden no coincidir con los mostrados en la imagen.

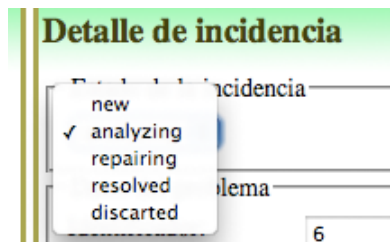


Figura 5-9: Detalle de estado de incidencias

De igual forma se configurarán la importancia y la clasificación de las incidencias, se presentará un menú desplegable similar al anterior dentro del apartado "Datos del problema".

Los datos del usuario se referirán al experto creador de la incidencia (Figura 5-10) o a los datos que el usuario haya indicando al enviar este (Figura 5-11), dependiendo del origen de la incidencia. A continuación se muestran los dos casos:

Incidencia creada por usuario experto:

Figura 5-10: Detalle de creador de incidencia (creador experto)

Incidencia creada por usuario del sitio web:

Figura 5-11: Detalle de creador de incidencia (acceso público)

Se podrá incluir un nuevo comentario sobre el proceso de resolución de la incidencia haciendo uso del área de texto con la etiqueta "Nuevo comentario", mostrada en la Figura 5-12. Los comentarios enviados se incluirán sobre este control como el que aparece en la siguiente imagen, indicando el usuario que lo realizó, la fecha y la hora del comentario y el texto especificado.

Figura 5-12: Detalle de comentarios de la incidencia

5.5. Creación de una nueva incidencia

Al pulsar sobre el enlace “Creación de nueva incidencia” del menú principal de la aplicación se accederá al módulo de creación de incidencias.

El módulo de creación de nuevas incidencias permite la inclusión en el sistema de incidencias por parte de los usuarios expertos, y ya que estas incidencias no se producen en base a la navegación sino a la revisión experta los campos del formulario son distintos, limitándose a los relativos a la incidencia.

El usuario rellenará los datos deseados en el interfaz de creación de nuevas incidencias (Figura 5-13), siempre proporcionando una dirección URL válida y perteneciente a alguno de los sitios web monitorizados, así como una descripción, ya que ambos campos son obligatorios. Finalmente se deberá pulsar el botón “Enviar” para mandar la incidencia al servidor y que esta quede almacenada. El botón “Restaurar” eliminará los datos introducidos en el formulario.

SIGA
Sistema Integral para la Gestión de la Accesibilidad

Inicio > Creación de nueva incidencia

Identificado como admin.

- Inicio
- Gestión de incidencias
- Creación de nueva incidencia
- Alertas automáticas
- Base de conocimiento
- Administración
- Manual de usuario
- Cerrar sesión

Creación de nueva incidencia

Interfaz destinado a la creación de nuevas incidencias.

Indique los datos de la incidencia, la url y la descripción son de carácter obligatorio.

Datos del problema

Url: *

Descripción: *

Importancia: Baja

Datos de modo de acceso

Navegador:

Resolución:

Enviar Restaurar

*: Campos requeridos

Figura 5-13: Interfaz de creación de incidencias (usuario experto)

Una vez enviada la incidencia se mostrará un mensaje al usuario indicando que esta ha sido correctamente enviada como el de la Figura 5-14, en caso de producirse algún error estos se mostrarían sobre el formulario, indicando el lugar donde se han producido para que quedan ser identificados fácilmente.

Creación de nueva incidencia

Incidencia creada correctamente.

Figura 5-14: Mensaje de incidencia creada correctamente

5.6. Visualización de alertas automáticas de accesibilidad

Al pulsar sobre el enlace “Alertas automáticas” del menú principal de la aplicación se accederá al módulo de visualización de alertas automáticas.

El módulo de alertas automáticas es una parte esencial del sistema que permite a los usuarios con permiso de acceso a esta funcionalidad visualizar las alertas que el sistema ha recogido mediante la una exploración automática para todos los sitios web monitorizados, incluyendo en la base de datos los fallos detectados, que podrán ser de ayuda a los expertos en accesibilidad o a los desarrolladores a la hora de solucionar posibles errores de accesibilidad antes de que los usuarios experimenten problemas debido a ellos y envíen incidencias y apoyando a las personas encargadas de las revisiones de accesibilidad de la web.

Las alertas de accesibilidad se clasificarán en primer lugar por el sitio web al que pertenecen y a continuación por el punto de verificación con el que están relacionadas. Por lo tanto la navegación dentro de este módulo se realizará según el esquema de la Figura 5-15.

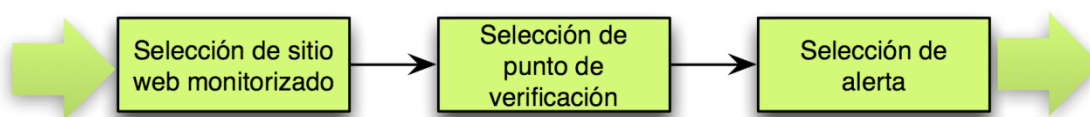


Figura 5-15: Proceso de visualización de alertas automáticas

Cada una de las cajas representa una acción del usuario, la cual se corresponderá a un interfaz de la aplicación detallado a continuación. Las flechas grandes se corresponden a la entrada y la salida del proceso.

A continuación se muestra el interfaz de selección de sitio web, detallado en la Figura 5-16. En el se indican las direcciones URL de estos y el estado de funcionamiento de la revisión automática, en el caso de que se mostrara el mensaje “Fallo” en alguno de los sitios web se deberá contactar con el administrador del sistema, ya que las alertas para este podrían no estar actualizadas.

Para visualizar los puntos de verificación se deberá pulsar sobre el botón “Ver checkpoints” del sitio deseado.

SIGA
Sistema Integral para la Gestión de la Accesibilidad

Inicio > Alertas automáticas

Identificado como admin.

- Inicio
- Gestión de incidencias
- Creación de nueva incidencia
- Alertas automáticas
- Base de conocimiento
- Administración
- Manual de usuario
- Cerrar sesión

Alertas automáticas

Las alertas automáticas se actualizan de forma autónoma gracias al analizador integrado en el sistema.

Lista de sitios configurados

Las alertas automáticas se clasifican en función de los sitios a los que pertenece, por favor seleccione el sitio deseado. A continuación se muestran los sitios configurados para su monitorización:

Para ver un detalle de los errores de accesibilidad pulse sobre el botón "Ver checkpoints" del sitio que desee.

Url del sitio	Sitio en línea	Acciones
http://www.uc3m.es	Ok	Ver checkpoints
http://www.ya.com	Ok	Ver checkpoints
http://www.telefonica.es	Ok	Ver checkpoints
http://www.google.com	Ok	Ver checkpoints

Figura 5-16: Interfaz de selección de sitio web monitorizado

A continuación se muestra el interfaz de selección de punto de verificación (Figura 5-17), el cual se accederá después de seleccionar el sitio web del que se desean ver las alertas. En él se mostrarán los puntos de verificación para los cuales existe alguna alerta de accesibilidad ordenados por su identificador. Se indica asimismo la importancia de este punto así como las recomendaciones asociadas. Para visualizar las alertas se deberá pulsar sobre el botón "Detalle" del punto de verificación deseado.

Nota: El sistema analiza los sitios web en base a las pautas de WCAG 1.0. En futuras versiones se implantará soporte para realizar este análisis en base a la versión 2.0 de WCAG o a la norma UNE aplicable en España.

SIGA
Sistema Integral para la Gestión de la Accesibilidad

Inicio > Alertas automáticas > Listado de puntos de verificación

Identificado como admin.

- Inicio
- Gestión de incidencias
- Creación de nueva incidencia
- Alertas automáticas
- Base de conocimiento
- Administración
- Manual de usuario
- Cerrar sesión

Alertas automáticas

Las alertas automáticas se actualizan de forma autónoma gracias al analizador integrado en el sistema.

Listado de puntos de verificación

Punto de comprobación	Importancia	Recomendaciones	Acciones
3.2	Media	Cree documentos que se ciñan a las gramáticas formales publicadas. (Prioridad 2)	Detalle
3.4	Media	Use unidades relativas en vez de absolutas en los valores de los atributos del lenguaje de marcado y en los valores de las propiedades de las hojas de estilo. (...)	Detalle
4.3	Baja	Identifique el lenguaje natural principal de un documento. (Prioridad 3)	Detalle
9.5	Baja	Proporcione atajos de teclado para los enlaces importantes (incluyendo los de los mapas de imagen de tipo cliente), controles de formulario y grupos de control...	Detalle

Figura 5-17: Interfaz de selección de punto de verificación

El interfaz de selección de alertas automáticas mostrará para el sitio web y el punto de verificación seleccionados un listado con las alertas automáticas encontradas (Figura 5-18), indicando tanto la dirección URL donde se ha encontrado, como un nombre identificativo para el error. Se deberá pulsar sobre el botón "Detalle" para ver todos los datos de las alertas.



Figura 5-18: Interfaz de selección de alerta automática

El interfaz de visualización de alertas automáticas (Figura 5-19) muestra todos los datos recogidos sobre estas, incluyendo los siguientes:

- Identificador univoco de la alerta utilizado por el sistema.
- Sitio, el cual indica el sitio web monitorizado al cual pertenece la incidencia.
- URL, muestra la ruta completa de la dirección URL donde se ha encontrado la alerta, la cual estará contenida dentro del sitio web.
- Checkpoint, que indica el punto de verificación al que pertenece la alerta, normalmente estos están identificados con una serie de 2 ó 3 números separados con puntos, lo que los clasifican en una estructura arbórea.
- Importancia, identifica la importancia del punto de verificación, esta normalmente está definida en 3 niveles, "Alta", "Media" y "Baja", que se corresponderían con los niveles de WCAG.
- Recomendaciones, son las recomendaciones generales que se especifican para este punto de verificación. Estas recomendaciones pueden ayudar a comprender el significado del punto de verificación y ayudar a la resolución del problema.
- Nombre del error, nombre identificativo de la alerta encontrada.
- Descripción, que indica detalladamente los fallos encontrados en el análisis automático.
- Como reparar, explica alguna de las posibles soluciones que se pueden aplicar para solucionar este problema, así como detalles de implementación u otra información relevante y de ayuda.

El botón "Ok" al final de la página cerrará esta alerta y regresará al interfaz de visualización de alertas.

Nota: Debido a la naturaleza automática de la obtención de las incidencias no se permite editar la información contenida en estas, ya que la nueva información sería borrada por el sistema de forma periódica.

Nota: Al realizar cambios sobre un sitio web monitorizado deberá esperar un tiempo de hasta 24 horas para que esta información se actualice.

Identificado como admin.

- Inicio
- Gestión de incidencias
- Creación de nueva incidencia
- Alertas automáticas
- Base de conocimiento
- Administración
- Manual de usuario
- Cerrar sesión

Alertas automáticas

Las alertas automáticas se actualizan de forma autónoma gracias al analizador integrado en el sistema.

Detalle de la alerta

Datos del problema

Identificador:	124549
Sitio:	http://www.ya.com
Url:	http://www.ya.com/chats
Checkpoint:	6.3
Importancia:	Alta
Recomendaciones:	Asegúrese de que las páginas pueden seguir siendo usadas cuando los scripts, applets u otros objetos de programación se desconectan o no son soportados. Si esto no es posible, proporcione información equivalente en

Nombre del error: Enlaces

Descripción: Hay 5 enlaces que se activan mediante scripts.

Como reparar: Se debe comprobar si el contenido del atributo "href" en los enlaces es un recurso válido, es decir, si se indica una dirección (<acronym title="Uniform Resource Identifier" xml:lang="en" lang="en">URI</acronym>) a la que se

Ok

© 2010 Adrian Lopez

W3C XHTML 1.0

W3C WAI-AAA WCAG 1.0

Figura 5-19: Interfaz de detalle de alerta automática

5.7. Gestión de la base de conocimiento

Al pulsar sobre el enlace "Base de conocimiento" del menú principal de la aplicación se accederá al módulo de gestión de la base de conocimiento.

El módulo de gestión del conocimiento permite a los usuarios con los permisos adecuados para su manejo la consulta de los instancias de conocimiento en base a las clasificaciones presentadas así como crear nuevos artículos de conocimiento y añadirlas a cualquier clasificación que se desee, pudiendo ser consultados más adelante.

Las instancias de conocimiento se clasificarán según una estructura arbórea, la navegación dentro de este módulo se realizará de la forma indicada en la Figura 5-20:

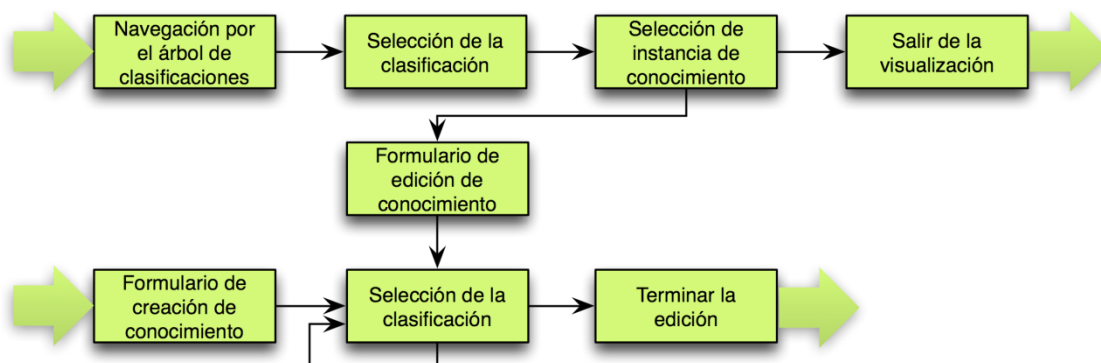


Figura 5-20: Procesos de gestión del conocimiento

Cada una de las cajas representa una acción del usuario, la cual se corresponderá a un interfaz de la aplicación detallado a continuación. Las flechas grandes se corresponden a la entrada y la salida del proceso.

En la Figura 5-21 se muestra el primer interfaz para la gestión de conocimiento, marcado con un ovalo se encuentran el botón de creación de nuevo conocimiento y el control de navegación por las categorías, el cual se presenta en forma de árbol, para desplegar este árbol será necesario pulsar sobre el enlace en forma de cruz (identificado mediante el texto alternativo "expandir" y para replegarlo en el símbolo en forma de guión que aparecerá al expandirlo (identificado mediante el texto alternativo "contraer"); para listar los artículos contenidos en una elementos de la clasificación se deberá pulsar sobre el botón que contiene el texto.

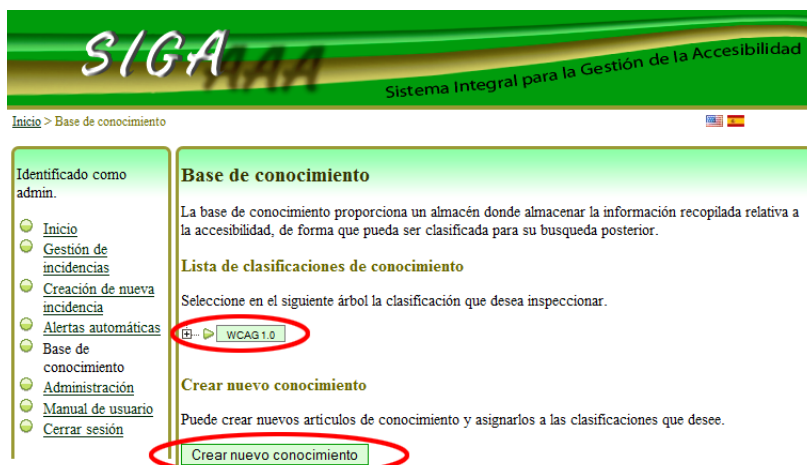


Figura 5-21: Interfaz de gestión del conocimiento

Se muestra a continuación un detalle del árbol expandido:

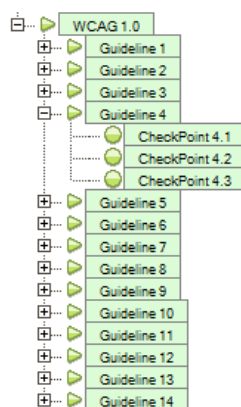


Figura 5-22: Detalle del árbol de categorías

5.7.1. Visualización de artículos de conocimiento

Una vez seleccionada una de las clasificaciones se mostrará el nombre y la descripción de esta, así como un listado de los artículos de conocimiento pertenecientes a esta, como se muestra en la Figura 5-23.

Base de conocimiento

La base de conocimiento proporciona un almacén donde almacenar la información recopilada relativa a la accesibilidad, de forma que pueda ser clasificada para su búsqueda posterior.

Conocimiento de Guideline 3

Descripción
Use markup and style sheets and do so properly.

Listado de conocimiento

Nombre del conocimiento	Descripción	Acciones
Prueba de conocimiento	Se presenta el contenido deseado	Detalle

Figura 5-23: Interfaz de listado de artículos de conocimiento por categorías

Al pulsar sobre el botón “Detalle” del artículo de conocimiento deseado se mostrará la totalidad de su información como se muestra en la Figura 5-24. En este interfaz se puede editar cualquiera de los campos y guardar el contenido mediante el botón “Enviar”, asimismo se puede recuperar el contenido original mediante el botón “Restaurar”.

Nota: La selección del elemento “Borrar clasificaciones actuales” eliminará todas las clasificaciones existentes para este artículo. Independientemente de si se selecciona o no esta caja al enviar los datos se le pedirá al usuario que seleccione las clasificaciones deseadas. Las nuevas clasificaciones seleccionadas se añadirán a las existentes en caso de que no se haya seleccionado su borrado.

SIGA MAA
Sistema Integral para la Gestión de la Accesibilidad

Inicio > Base de conocimiento > Detalle de conocimiento

Identificado como admin.

- Inicio
- Gestión de incidencias
- Creación de nueva incidencia
- Alertas automáticas
- Base de conocimiento
- Administración
- Manual de usuario
- Cerrar sesión

Base de conocimiento

La base de conocimiento proporciona un almacén donde almacenar la información recopilada relativa a la accesibilidad, de forma que pueda ser clasificada para su búsqueda posterior.

Datos del artículo de conocimiento

Identificador: 1

Nombre: Prueba de conocimiento

Contenido: Se presenta el contenido deseado

☐ Borrar clasificaciones actuales

Modificar Restaurar

Figura 5-24: Interfaz de visualización y edición de artículos de conocimiento

Una vez pulsado el botón “Enviar” se presentará el siguiente interfaz (Figura 5-25) de selección de categorías:

SIGA MAA
Sistema Integral para la Gestión de la Accesibilidad

Inicio > Base de conocimiento > Crear nuevo conocimiento

Identificado como admin.

- Inicio
- Gestión de incidencias
- Creación de nueva incidencia
- Alertas automáticas
- Base de conocimiento
- Administración
- Manual de usuario
- Cerrar sesión

Base de conocimiento

La base de conocimiento proporciona un almacén donde almacenar la información recopilada relativa a la accesibilidad, de forma que pueda ser clasificada para su búsqueda posterior.

Seleccionar la clasificación del conocimiento

Para el artículo de contenido creado anteriormente, seleccione las clasificaciones a las que estará asociado.

WCAG 1.0

- Guideline 1
- Guideline 2
- Guideline 3
- Guideline 4
- Guideline 5
- Guideline 6
- Guideline 7
- Guideline 8
- Guideline 9
- Guideline 10
- Guideline 11
- Guideline 12
- Guideline 13
- Guideline 14

Una vez haya finalizado la definición de clasificaciones pulse el botón "Terminar".

Terminar

Figura 5-25: Interfaz de selección de categorías asociadas

Para cada una de las categorías deseada se deberá pulsar sobre su botón (el sistema es idéntico al de navegación por la categorías). Una vez seleccionadas todas se pulsará sobre el botón “Terminar”, finalizando la edición.

5.7.2. Creación de nuevos artículos de conocimiento

Cuando se accede a la página de creación de nuevos artículos de conocimiento se presenta un interfaz similar al de edición de artículos (Figura 5-24 y Figura 5-25), consulte la sección anterior para conocer como se deben crear y editar estos.

Nota: Para que un artículo pueda ser consultado este debe agregarse al menos a una de las categorías.

5.8. Administración del sistema

Al pulsar sobre el enlace “Administración” del menú principal de la aplicación se accederá al módulo de administración del sistema.

El módulo de administración del sistema permite a un usuario con los permisos adecuados configurar los diferentes aspectos de la aplicación para ajustarlo a las necesidades de los usuarios. Las funciones presentadas son las siguientes, las cuales se ofrecen en la interfaz de la Figura 5-26:

- Administración de roles y permisos: Permite la creación, modificación y borrado de los roles del sistema, pudiendo especificar los permisos asociados a cada uno de estos. Los roles permiten la creación de grupos de usuarios con los mismos permisos que estarán dedicados a las mismas labores.
- Administración de usuarios y roles: Permite la creación, modificación y borrado de los usuarios del sistema, estableciendo los roles que toma cada uno de ellos. Los usuarios permiten el acceso diferenciado a las personas que hacen uso de la aplicación, normalmente cada una de los usuarios dispondrá solo de uno de los roles, sin embargo cuando el número de usuarios es reducido puede que un mismo usuario tenga que desempeñar varios de estos o incluso todos, por lo que la aplicación permite total libertad en este aspecto.
- Administración de estados de incidencias: Permite la inclusión y borrado de los estados permitidos para las incidencias del sistema. Estos estados dependerán del proceso definido para la resolución de las incidencias. Sin embargo el estado “New” (nueva) es obligatorio y será el que tomen todas las incidencias al ser creadas.
- Administración de categorías: Permite añadir y eliminar las categorías posibles para las incidencias del sistema.
- Configuración de los sitios monitorizados: Permite añadir y eliminar los sitios que el sistema está encargado de monitorizar. Además permite ver si el estado de estos.



Figura 5-26: Interfaz de administración

5.8.1. Sistema de permisos

El diagrama de la Figura 5-27 ilustra el sistema de permisos de la aplicación.

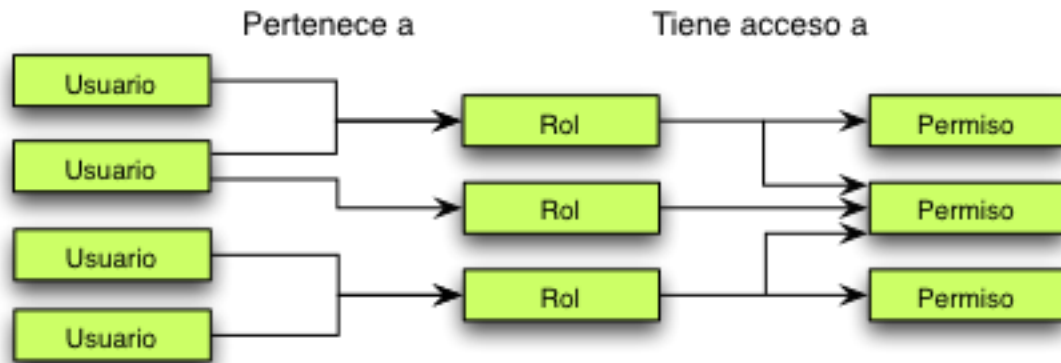


Figura 5-27: Diagrama de estructura de permisos

Como se puede apreciar, cada uno de los usuarios pertenece a uno o varios roles y cada rol contendrá un determinado número de usuarios, que usualmente será mayor que uno. A su vez cada uno de los roles tiene acceso a un conjunto de permisos, que son los que definen las partes de la aplicación a las que tendrá acceso.

El sistema de roles y permisos facilita las labores de administración cuando se requiere configurar un número relativamente elevado de usuarios aumentando la flexibilidad de la gestión, ya que los permisos se asignan a los roles, evitando tener que configurar los permisos para cada uno de los usuarios por separado. Para entornos con pocos usuarios puede que no se obtenga beneficio de este sistema, sin embargo no supondrá apenas esfuerzo adicional la configuración de los roles.

5.8.2. Administración de roles y permisos

En la página de gestión de roles (Figura 5-28) se muestra un listado de los roles actuales en la parte superior, el cual presenta tanto el identificador como el nombre y la descripción de cada uno de estos, asimismo para cada uno de ellos existen los botones "Modificar" y "Eliminar" que permitirán realizar sendas acciones.

Nota: No se podrá eliminar el rol del propio usuario, ya que esto podría provocar que la aplicación quedara sin ningún usuario administrador, lo cual no tendría vuelta atrás.

La parte inferior de la página presenta un formulario para la creación de un nuevo rol, en este se deberá especificar un nombre del rol (el cual será obligatorio y no podrá estar ya en uso) y una descripción, así como los permisos que se le deseen asignar, pulsando sobre el botón "Crear" para completar la creación del nuevo rol.



Figura 5-28: Interfaz de administración de roles y permisos

Si se pulsa sobre el botón “Modificar” se accederá a un nuevo interfaz donde se podrán editar los datos del rol seleccionado (Figura 5-29). A continuación se muestra un ejemplo:

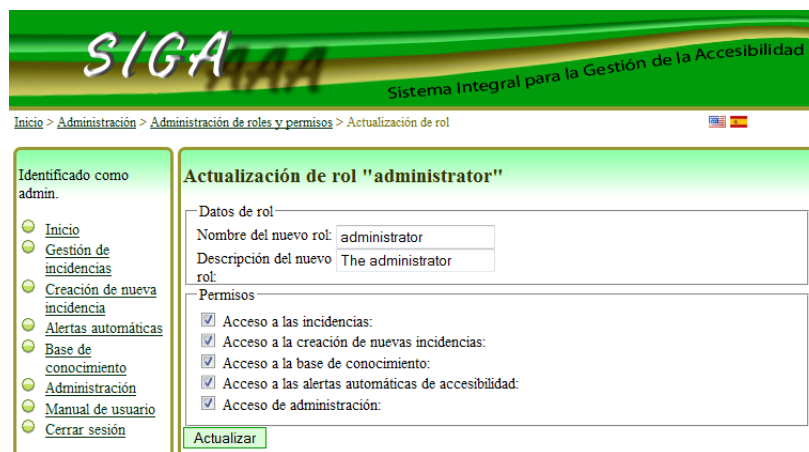


Figura 5-29: Interfaz de modificación de roles

5.8.3. Administración de usuarios y roles

En la página de gestión de usuarios (Figura 5-30) se muestra un listado de los usuarios actuales en la parte superior, el cual presenta tanto el identificador como el nombre y la descripción de cada uno de estos, asimismo para cada uno de ellos existen los botones “Modificar” y “Eliminar” que permitirán realizar sendas acciones.

Nota: No se podrá eliminar el propio usuario, ya que esto podría provocar que la aplicación quedara sin ningún usuario administrador, lo cual no tendría vuelta atrás.

La parte inferior de la página presenta un formulario para la creación de un nuevo usuario, en este se deberá especificar un nombre de usuario (el cual será obligatorio y no podrá estar ya en uso) y una contraseña (que no podrá ser vacía), pulsando sobre el botón “Crear” para completar la creación del nuevo usuario.

Identificado como admin.

- Inicio
- Gestión de incidencias
- Creación de nueva incidencia
- Alertas automáticas
- Base de conocimiento
- Administración
- Manual de usuario
- Cerrar sesión

Administración de usuarios y roles

Este interfaz permite la creación, modificación y borrado de los usuarios del sistema, pudiendose asignar a cada uno los roles deseados.

Listado de los roles actuales:

Id de usuario	Nombre de usuario	Rol	Acciones
1	admin	administrator	Modificar Eliminar
2	analyst	analyst	Modificar Eliminar
3	programmer	programmer	Modificar Eliminar
4	inspector	inspector	Modificar Eliminar
5	AccessibilityExpert	AccessibilityExpert	Modificar Eliminar

Crear nuevo usuario:

Nombre de usuario:

Contraseña:

Figura 5-30: Interfaz de administración de usuarios y roles

Si se pulsa sobre el botón “Modificar” se accederá a un nuevo interfaz (Figura 5-31) donde se podrán editar los datos del usuario seleccionado así como asignarle a este los roles del sistema que se deseen. A continuación se muestra un ejemplo:

Identificado como admin.

- Inicio
- Gestión de incidencias
- Creación de nueva incidencia
- Alertas automáticas
- Base de conocimiento
- Administración
- Manual de usuario
- Cerrar sesión

Actualización del usuario "admin"

Este interfaz permite incluir nuevos datos para el usuario.

Datos de usuario

Nombre de usuario:

Contraseña:

Roles

- ☐ analyst
- ☐ inspector
- ☐ AccessibilityExpert
- ☐ programmer
- ☒ administrator

Figura 5-31: Interfaz de modificación de usuarios

5.8.4. Administración de estados de incidencias

En la página de gestión de los estados (Figura 5-32) de las incidencias se muestra un listado de los estados en la parte superior, el cual presenta tanto el identificador como el nombre y la descripción de cada uno, asimismo para cada uno de ellos existen los botones “Modificar acceso” y “Eliminar” que permitirán realizar sendas acciones.

La parte inferior de la página presenta un formulario para la creación de un nuevo estado, en este se deberá especificar un nombre de usuario (el cual es obligatorio y no podrá estar ya en uso) y una descripción, pulsando posteriormente en el botón “Crear”.

SIGA
Sistema Integral para la Gestión de la Accesibilidad

Inicio > Administración > Administración de estados de incidencias

Identificado como admin.

- Inicio
- Gestión de incidencias
- Creación de nueva incidencia
- Alertas automáticas
- Base de conocimiento
- Administración
- Manual de usuario
- Cerrar sesión

Administración de estados de incidencias

Este interfaz permite la creación, modificación y borrado de los estados de las incidencias del sistema.

Listado de los estados de incidencia actuales:

Identificador de estado	Nombre de estado	Descripción	Acciones
1	new		Modificar Acceso Eliminar
2	analyzing		Modificar Acceso Eliminar
3	repairing		Modificar Acceso Eliminar
4	resolved		Modificar Acceso Eliminar
5	discarted		Modificar Acceso Eliminar

Crear nuevo estado de incidencia:

Nombre del nuevo estado:

Descripción del nuevo estado:

[Crear](#)

Figura 5-32: Interfaz de administración de estados de incidencias

5.8.5. Administración de categorías

En la página de gestión de categorías (Figura 5-33) se muestra un listado de las categorías actuales en la parte superior, el cual presenta tanto el identificador como el nombre y la clasificación de cada una de estas, asimismo para cada uno de ellas existen el botón “Eliminar” que permitirán realizar esta acción.

Nota: No se podrá eliminar una clasificación que tenga clasificaciones hijas.

La parte inferior de la página presenta un formulario para la creación de una nueva categoría, en este se deberá especificar su nombre (el cual será obligatorio y no podrá estar ya en uso) y seleccionar la categoría de la que cuelga en el árbol, pulsando sobre el botón “Crear” para completar la creación.

Administración de categorías

Este interfaz permite la creación y borrado de las clasificaciones de las incidencias.

Listado de las clasificaciones de incidencias actuales:

Identificador de la clasificación	Nombre de las clasificación	Clasificador superior	Acciones
1	general	general	Eliminar
2	clasificacion 1	general	Eliminar
3	clasificacion hija	clasificacion 1	Eliminar
4	otra clasificacion	general	Eliminar

Nueva clasificación de incidencias:

Nombre de la nueva clasificación:

Clasificación superior de la nueva clasificación:

Figura 5-33: Interfaz de administración de categorías

5.8.6. Administración de sitios monitorizados

En la página de gestión de sitio monitorizados (Figura 5-34) se muestra un listado de estos en la parte superior, el cual presenta tanto el identificador como la dirección URL base y el estado para cada uno, además de el botón “Eliminar” que permitirá realizar esta acción.

La parte inferior de la página presenta un formulario para la configuración de un nuevo sitio, en este se deberá especificar su dirección base (el cual será obligatorio y no podrá estar ya en uso), pulsando sobre el botón “Crear” para completar el proceso.

Configuración de los sitios monitorizados

Este interfaz permite la creación y borrado de los sitios monitorizados por el sistema, permitiendo tanto el análisis automático como la inclusión de incidencias por parte de los usuarios.

Listado de los sitios monitorizados actualmente:

Id del sitio	URL base	Estado	Acciones
1	http://www.uc3m.es	Ok	Eliminar
2	http://www.ya.com	Ok	Eliminar
3	http://www.telefonica.es	Ok	Eliminar
4	http://www.google.com	Ok	Eliminar

Monitorizar nuevo sitio web:

Url base del nuevo sitio:

Figura 5-34: Interfaz de administración de sitios monitorizados

Capítulo 6.

Validación

En este capítulo se muestra la validación de la aplicación SIGA, Sistema Integral para la Gestión de la Accesibilidad. En esta validación se busca comprobar que se cumplen los siguientes objetivos:

- Objetivo A: La aplicación cumple con todos los requisitos de usuario y de software definidos.
- Objetivo B: La aplicación es usable y la curva de aprendizaje necesaria para su uso completo es ligera.

Para la comprobación del objetivo A se ha realizado una batería de pruebas manuales destinadas a comprobar que todos los puntos definidos en los requisitos quedan cubiertos en el uso de la aplicación dentro del entorno operacional indicado. En esta batería de pruebas se pueden destacar de los siguientes tipos:

- Pruebas de funcionalidad: Se ha comprobado que toda la funcionalidad especificada en los requisitos queda cubierta en la aplicación.
- Pruebas de entrada de usuario: Se ha comprobado que la salida del sistema es correcta para todos los campos para cualquier entrada por parte del usuario, aunque esta no siga el formato esperado, no apareciendo errores en la aplicación.
- Pruebas de integridad de los datos: Se ha comprobado que las entradas de usuario, sean erróneas o malintencionadas no ponen en peligro la integridad de los datos de la aplicación.
- Pruebas de carga: Se ha comprobado que la aplicación responde correctamente ante situaciones de alta carga del sistema, produciéndose los retrasos esperados en las peticiones por las limitaciones hardware y de red pero sin que se produzcan errores.
- Pruebas de estabilidad: Se han realizado pruebas de estabilidad del sistema, comprobando que este no se vuelve inestable tras un periodo de ejecución de varias semanas.

Para la comprobación del objetivo B, adoptando un enfoque de diseño centrado en el usuario (DCU) se ha diseñado y llevado a cabo una prueba con usuarios. Para ello, se ha seleccionado un grupo de personas con distintos niveles de experiencia en accesibilidad y usabilidad. A cada una de las personas que han participado en esta prueba se les ha proporcionado:

- Una presentación del objetivo y funcionalidad básica de la aplicación como introducción.
- El manual de usuario de la aplicación como referencia y ayuda para el ejercicio.
- Un ejercicio con algunas de las tareas más comunes a realizar en la aplicación (ver Anexo F).
- Un cuestionario que el usuario deberá rellenar en base a su experiencia con la aplicación (ver Anexo G).

Esta prueba pretende medir la satisfacción del objetivo B en base a los resultados obtenidos en los cuestionarios realizados por los usuarios. A continuación se muestra el análisis de los resultados:

6.1. Tiempo empleado

A continuación se analiza el tiempo utilizado por los usuarios que han realizado la prueba para completar las tareas especificadas en el guión, este análisis puede servir como una medición de la usabilidad de la aplicación, basada en obtener resultados de la satisfacción del usuario y facilidad de uso.

El tiempo medio estimado para completar las tareas se estableció en 15 minutos, y como se observa en las gráficas siguientes sobre el tiempo real utilizado la estimación ha sido bastante acertada:

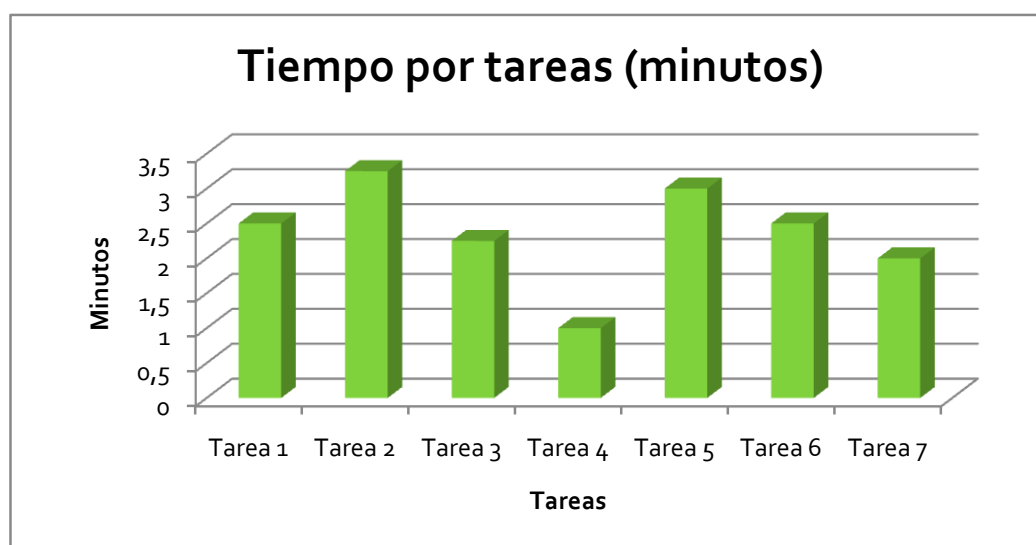


Figura 6-1: Gráfico de tiempo empleado por tareas

Cómo se puede apreciar, los tiempos medios para la realización de las tareas solicitadas ha sido muy reducido, siendo la media más elevada 3 minutos. Estos tiempos son satisfactorios e indican que las tareas se han desarrollado sin problemas mayores por parte de los usuarios. En la Figura 6-2 se puede apreciar el tiempo por tareas para cada uno de los usuarios que ha realizado la prueba.

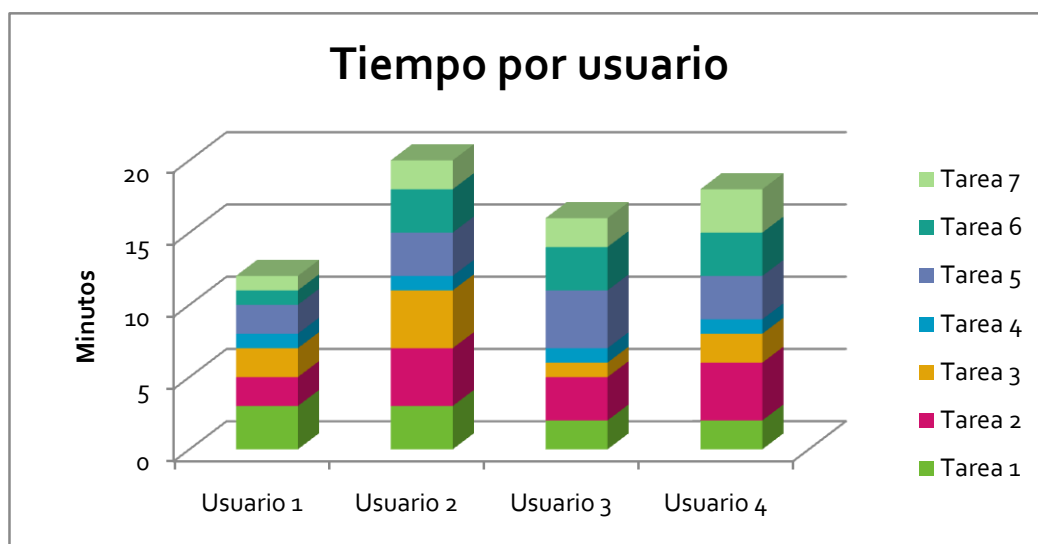


Figura 6-2: Gráfico de tiempo empleado por usuario en cada tarea

6.2. Valoración en el cuestionario

A continuación se muestran las gráficas con las valoraciones medias a las preguntas realizadas en el cuestionario. La escala sobre la que se ha valorado toma los valores desde -2 a 2, siendo 2 el valor más positivo.

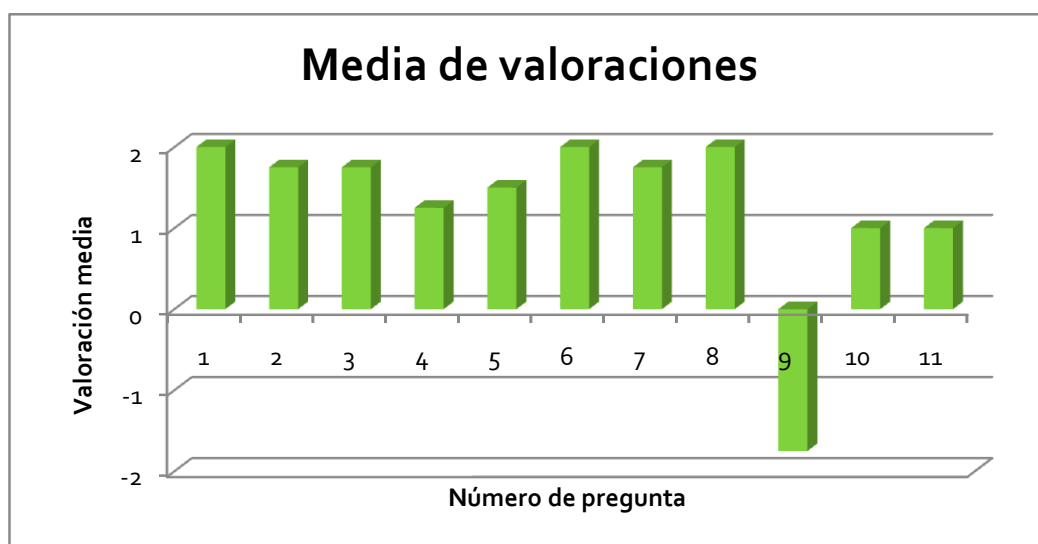


Figura 6-3: Media de valoraciones en las preguntas del cuestionario

Como se puede apreciar los resultados para todas las preguntas toman unos valores elevados, siendo la media igual o superior a 1 salvo para la pregunta 9 que toma valores negativos. Esto es debido a que una respuesta positiva implicaba que hubiera habido necesidad de uso del manual de usuario por parte de las personas involucradas en la prueba, sin embargo todos indicaron que no fue necesario el uso del manual en absoluto (un valor de -2), salvo una de las

personas que indicó que casi no fue necesario (un valor de -1). Con respecto a la pregunta número 10, esta ha sido respondida para el caso del usuario que valoró con -1 la pregunta 9. La pregunta número 11, sobre la estructura del manual de usuario, que ha recibido también una valoración menor ha recibido un *feedback* que ha permitido mejorar dicho manual.



Figura 6-4: Varianza de valoraciones en las preguntas del cuestionario

Se puede comprobar cómo la varianza en las respuestas ha sido muy baja, no superando salvo para la pregunta 11 los 3,35 puntos, lo que indica que las respuestas han sido bastante uniformes.

6.3. Observaciones del cuestionario

Cabe destacar que en las observaciones varias personas indicaron la conveniencia de incluir un índice en el manual de usuario, tarea que se ha realizado, mejorando su facilidad de uso y claridad.

6.4. Conclusiones

Los resultados obtenidos tanto en los tiempos de realización de las pruebas, como en las respuestas al cuestionario y en los comentarios de los usuarios tras la realización del ejercicio han sido satisfactorios y permiten afirmar que el sistema cumple con los objetivos de funcionalidad y calidad marcados.

Capítulo 7. Conclusiones y Líneas futuras

7.1. Conclusiones

Tanto el desarrollo del proyecto, incluyendo todas sus fases, como la creación de la presente memoria ha sido satisfactoria, la aplicación ha cumplido y posiblemente superado las expectativas iniciales.

La aplicación SIGA, Sistema Integral para la Gestión de la Accesibilidad permite a un equipo de desarrolladores web, analistas y expertos en accesibilidad trabajar conjuntamente en el mantenimiento de la accesibilidad de un sitio web de una forma ágil y sencilla, facilitando la labor de todos los miembros. Asimismo puede ayudar en este labor a equipos unipersonales por medio de la monitorización y el modulo de gestión de incidencias. Los requisitos que se fijaron en el análisis del sistema se han visto cubiertos y se han proporcionado las funcionalidades de gestión de incidencias de usuarios, visualización de alertas automáticas y gestión del conocimiento tal y como quedó establecido en los objetivos.

Se ha conseguido desarrollar una aplicación de sencillo manejo, que requiere un esfuerzo mínimo por parte del usuario y una curva de aprendizaje muy asequible, presentando además un interfaz simple y agradable que maximiza la usabilidad y la accesibilidad. Este objetivo se ha conseguido en parte debido al uso de un interfaz web, con el cual la mayoría de los usuarios de ordenadores personales están familiarizados.

Tal como se buscó desde un inicio, la aplicación está preparada para ser utilizada en cualquier organización o particular que tenga interés en mantener un alto nivel de accesibilidad en sus páginas web, así como empresas dedicadas a la creación de páginas web que ofrezcan un servicio de mantenimiento y gestión de los problemas de los usuarios en la web. Por todo esto, se puede concluir que el panorama web actual podría obtener notables beneficios del uso de esta aplicación u otras similares, debido a la necesidad de procesos que aseguren la accesibilidad de los sitios web a lo largo de todo su ciclo de vida y evitando que la accesibilidad conseguida en el momento de creación se desvanezca con el tiempo debido a la agregación de contenido no accesible o de elementos web nuevos que no han sido diseñados con las pautas de accesibilidad en mente.

A nivel personal tengo que destacar que el diseño e implementación de esta aplicación ha sido un reto, no solamente por la envergadura de este proyecto para afrontarlo en solitario, sino por el uso de dos tecnologías como son Struts2 e Hibernate, las cuales desconocía por completo al inicio del proyecto. Si bien podría parecer que el uso de estas tecnologías facilita enormemente el desarrollo, y eso será cierto para cualquier persona con suficiente experiencia en ellas, sin embargo cada una de las tecnologías indicadas tiene una gran complejidad y una curva de aprendizaje de considerable dificultad, que se ve incrementada al hacer uso de las dos al mismo tiempo, lo que ha generado debido a mi desconocimiento varios cambios en el diseño inicial e infinidad de complicaciones en la implementación. Sin embargo es de agradecer tanto la excelente documentación que he podido encontrar sobre estas tecnologías como la ayuda que proporciona la comunidad en los foros de desarrolladores, sin este apoyo el esfuerzo hubiera sido muchísimo mayor y el resultado podría presentar una calidad inferior. El balance final del proyecto ha sido muy positivo, ya que la satisfacción y la experiencia que me ha proporcionado en el desarrollo de aplicaciones web supera con creces al esfuerzo realizado. Volviendo la vista atrás impresiona ver como mediante el trabajo, una sola idea inicial evoluciona y se desarrolla paso a paso hasta convertirse en una herramienta que cumple con los objetivos marcados en el primer paso de su evolución.

7.2. Líneas futuras

Debido a las limitaciones que se han de definir en la realización de un proyecto de fin de carrera, han quedado fuera del alcance de este un conjunto de funcionalidades que complementarían al conjunto actual del sistema. Se han clasificado estos aspectos funcionales según la necesidad y prioridad que tendrían de cara a la futura ampliación de funcionalidad, siendo estas clasificaciones en orden de mayor a menor prioridad: Muy convenientes, Convenientes y Opcionales; así como según el coste que supondría su implementación en horas de desarrollo (horas/persona). Los nuevos requisitos de las líneas futuras se detallan en las tablas LF01 a LF14. A continuación se detalla la nomenclatura utilizada así como estas líneas futuras:

- **Identificador:** formado por las siglas de "Línea futura" seguido de un número secuencial dado por orden de creación.
- **Descripción:** explicación formal de la línea futura.
- **Necesidad:** valoración de la importancia de implementación de la línea futura. La valoración sigue los valores:
 - Muy conveniente: la línea futura mejoraría de forma importante la funcionalidad del sistema para la mayor parte de los usuarios.
 - Convenientes: la línea futura mejoraría de forma importante la funcionalidad del sistema para algunos usuarios o mejoraría moderadamente la funcionalidad del sistema para la mayor parte de los usuarios.
 - Opcionales: la importancia de la línea futura no se puede enmarcar dentro de las anteriores, la implementación de esta tendrá un impacto limitado en la funcionalidad del sistema.
- **Coste:** estimación del coste en horas de desarrollo (horas/persona), incorporando análisis, diseño, implementación y pruebas.

- Alto: el coste de implementar esta funcionalidad se calcula superior a 50 horas.
- Medio: coste de implementar esta funcionalidad se calcula superior a las 15 horas e inferior a 50 horas.
- Bajo: coste de implementar esta funcionalidad se calcula inferior a 15 horas.

Tabla 7-1: LFo1

Identificador: LFo1
Descripción: Proporcionar una interfaz de gestión y modificación de las clasificaciones para las instancias de conocimiento. Actualmente estas modificaciones es necesario realizarlas directamente en la base de datos, lo que no supone ningún problema en el momento de despliegue de la aplicación, pero dificultaría la tarea de realizar modificaciones por parte de una persona con un perfil no técnico o sin conocimiento de manejo de bases de datos o de la sintaxis SQL necesaria.
Necesidad: Media
Coste: Bajo

Tabla 7-2: LFo2

Identificador: LFo2
Descripción: Proporcionar un mecanismos para buscar instancias de conocimiento en base a un campo de búsqueda, que pueda incluir opciones para buscar en el título, en el cuerpo del artículo o incluso por autor de este; y que devuelva al usuario una página con la relación de todos los resultados.
Necesidad: Media
Coste: Medio

Tabla 7-3: LFo3

Identificador: LFo3
Descripción: Proporcionar un mecanismos para buscar incidencias en base a un campo de búsqueda, que pueda incluir opciones para buscar por cualquiera de los campos de estas; y que devuelva al usuario una página con la relación de todos los resultados.
Necesidad: Baja
Coste: Medio

Tabla 7-4: LFo4

Identificador: LFo4
Descripción: Integrar un sistema de reconocimiento de lenguaje natural para clasificación automática de las incidencias creadas por usuarios.
Necesidad: Baja
Coste: Alto

Tabla 7-5: LFo5

Identificador: LFo5
<p>Descripción: Implementar un sistema de estadísticas que permita obtener datos de manera gráfica y resumida sobre los siguientes aspectos:</p> <ul style="list-style-type: none"> • Incidencias creadas en el sistema <ul style="list-style-type: none"> ○ Páginas web en las que se han producido ○ Problemas encontrados ○ Tiempo medio, mínimo y máximo de respuesta ○ Tiempo medio, mínimo y máximo de resolución ○ Porcentaje de incidencias resueltas satisfactoriamente ○ Desglose de datos por sitios • Alertas automáticas del sistema <ul style="list-style-type: none"> ○ Resumen por sitio de las incidencias automáticas encontradas ○ Resumen por página web de las incidencias automáticas encontradas ○ Resumen de los puntos de comprobación que han generado más alertas • Sitios web <ul style="list-style-type: none"> ○ Índice de calidad del sitio web, medido en función de las incidencias y alertas detectadas • Usuarios <ul style="list-style-type: none"> ○ Resumen de respuestas y acciones en el sistema por usuarios ○ Resumen de actividad de los usuarios
Necesidad: Baja
Coste: Alto

Tabla 7-6: LFo6

Identificador: LFo6
<p>Descripción: Proporcionar la opción de ignorar alertas automáticas en caso de que estas no se consideren relevantes por parte de los expertos de accesibilidad, por ejemplo puntos de comprobación de WCAG 1.0 que indican que se deben cumplir hasta que los agentes de usuario dejen de necesitarlo (marcados como <i>Until user agents</i>) y para los que actualmente se cumpla esto, y por lo tanto no sean ya necesarios.</p>
Necesidad: Media
Coste: Media

Tabla 7-7: LFo7

Identificador: LFo7
<p>Descripción: Almacenar las contraseñas de usuarios cifradas en la base de datos, de forma que un usuario que consiga acceso a esta no pueda recopilar estas contraseñas.</p>
Necesidad: Media
Coste: Bajo

Tabla 7-8: LFo8

Identificador: LFo8
<p>Descripción: Implantar un sistema de cifrado SSL en la página dedicada a la autenticación del usuario, evitando de esta forma en la medida de lo posible que el sistema sea vulnerable a</p>

un ataque *Man in the middle* y que un usuario malintencionado pueda hacerse con los datos de entrada a la aplicación.

Necesidad: Media

Coste: Medio

Tabla 7-9: LF09

Identificador: LF09

Descripción: Implantar un sistema de cifrado SSL en las páginas dedicada a la creación y gestión de incidencias, ya que estas pueden contener datos relativos a la salud de los usuarios además de datos que les identifican y por lo tanto esta información debería ser tratada con la máxima discreción posible. De esta forma se evitaría en la medida de lo posible que el sistema sea vulnerable a un ataque *Man in the middle* y que un usuario malintencionado pueda hacerse con los datos privados de los usuarios.

Necesidad: Media

Coste: Medio

Tabla 7-10: LF10

Identificador: LF10

Descripción: Implementar un sistema de memoria caché que sea capaz de almacenar los datos más accedidos, utilizando una política de reemplazo LRU o similar, acelerando el funcionamiento de la aplicación al no tener que realizar cada consulta a la base de datos. Sería necesario hacer un estudio de estrés para encontrar el algoritmo de reemplazo que más se ajuste a las necesidades, así como el tamaño de la caché más conveniente.

Necesidad: Baja

Coste: Medio

Tabla 7-11: LF11

Identificador: LF11

Descripción: Ampliar la funcionalidad del sistema de rastreo automático de páginas web, de forma que se haga uso de los ficheros "sitemap.xml" o "sitemap.html", así como poder especificar una ubicación de estos ficheros en la aplicación, de esta forma el seguimiento de la accesibilidad de las páginas web de un sitio será más limpio y óptimo.

Necesidad: Media

Coste: Medio

Tabla 7-12: LF12

Identificador: LF12

Descripción: Implementar funcionalidad que permita obtener un informe EARL en base de las alertas automáticas y las incidencias.

Necesidad: Media

Coste: Medio

Tabla 7-13: LF13

Identificador: LF13

Descripción: Incluir métricas que permitan medir la accesibilidad de un sitio web como un valor numérico específico, dando una puntuación a cada alerta e incidencia que se restarán de un máximo establecido.

Necesidad: Media

Coste: Medio

Tabla 7-14: LF14

Identificador: LF14

Descripción: Incluir soporte a análisis de páginas web que requieran autenticación en servidor o un proceso complejo para acceder a ellas. Se podrá definir los datos de entrada y/o el proceso para acceder a estas páginas para que puedan ser analizadas automáticamente.

Necesidad: Alta

Coste: Alto

Presupuesto

A continuación se muestra un diagrama de Gantt que representa la planificación establecida para el presente proyecto, dividida para cada una de sus fases.

Se indica para cada tarea, su nombre, el cual será identificativo de esta, la duración estimada en días y las fechas de inicio y fin que se han planificado. En la lado derecho se agrupan estos datos de forma gráfica y se indica además las personas implicadas en la realización de cada una de las tareas, con las siglas ALP (Adrián López Pérez) y LML (Lourdes Moreno López)

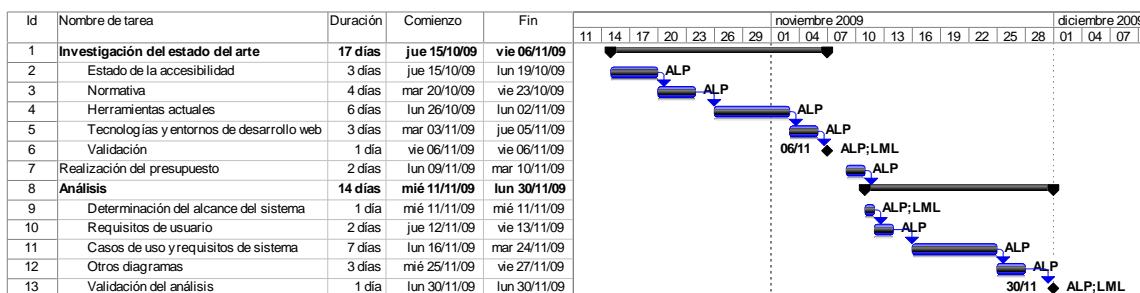


Figura 7-1: Diagrama de Gantt para las tareas Investigación del estado del arte, Realización del presupuesto y Análisis

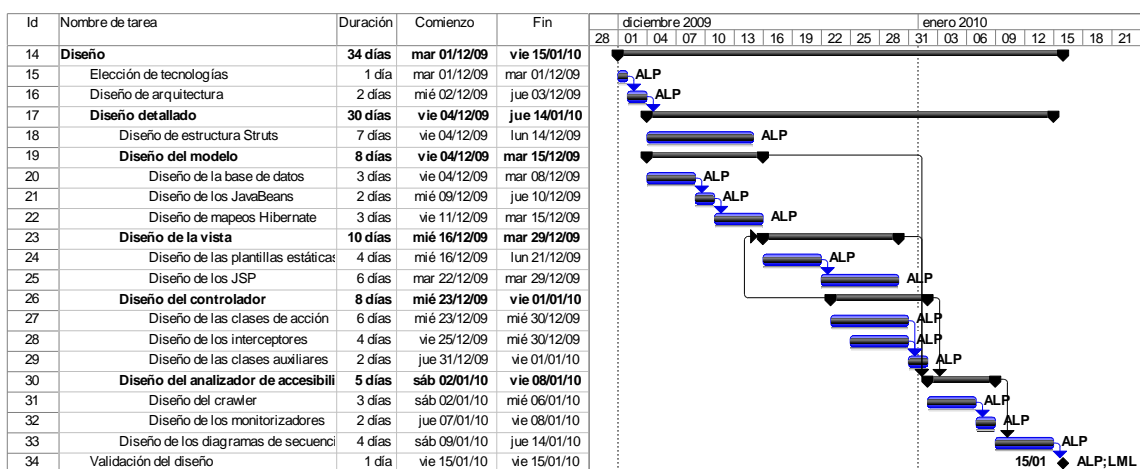


Figura 7-2: Diagrama de Gantt para la tarea Diseño

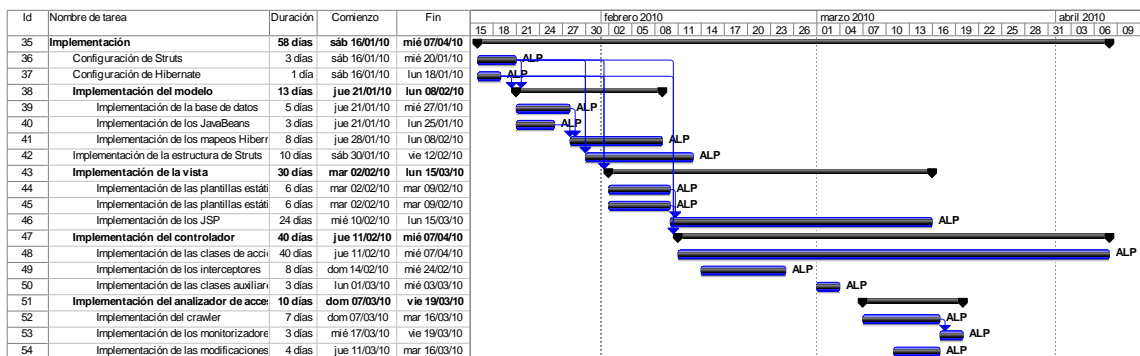


Figura 7-3: Diagrama de Gantt para la tarea Implementación

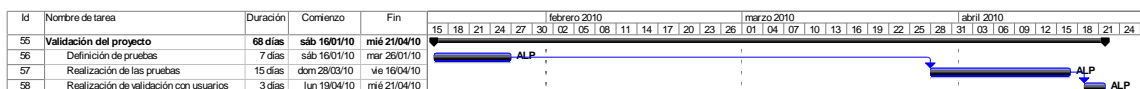


Figura 7-4: Diagrama de Gantt para la tarea Validación

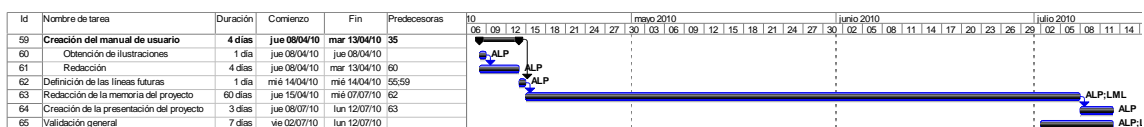


Figura 7-5: Diagrama de Gantt para las tareas Creación del manual de usuario, Definición de líneas futuras, Redacción de la memoria del proyecto, Creación de la presentación del proyecto y Validación general

A continuación se muestra el desglose de gastos del proyecto:



UNIVERSIDAD CARLOS III DE MADRID
Escuela Politécnica Superior

PRESUPUESTO DE PROYECTO

1.- Autor:

Adrián López Pérez

2.- Departamento:

Departamento de Informática

3.- Descripción del Proyecto:

- Título SIGA, Sistema Integral para la Gestión de la Accesibilidad

- Duración (meses) 9

Tasa de costes Indirectos: 20%

4.- Presupuesto total del Proyecto (valores en Euros):

22.309,01 Euros

5.- Desglose presupuestario (costes directos)

PERSONAL

Apellidos y nombre	N.I.F. (no rellenar - solo a título informativo)	Categoría	Dedicación personas mes ^{a)}	Coste persona mes	Coste (Euro)	Firma de conformidad
López Pérez, Adrián	00000000-X	Ingeniero	6,03	2.694,39	16.247,17	
Moreno López, Lourdes	00000000-X	Doctora	0,5	4.289,54	2.144,77	
					0,00	
					0,00	
					0,00	
Personas mes			6,53	Total	18.391,94	

EQUIPOS

Descripción	Coste (Euro)	% Uso dedicado proyecto	Dedicación (meses)	Periodo de depreciación	Coste imputable ^{d)}
Portatil Apple Macbook	1.500,00	100	9	60	225,00
PC Sobremesa	700,00	100	9	60	105,00
					0,00
					0,00
					0,00
					0,00
Total					330,00

SUBCONTRATACIÓN DE TAREAS

Descripción	Empresa	Coste imputable
Total		0,00

OTROS COSTES DIRECTOS DEL PROYECTO ^{d)}

Descripción	Empresa	Costes imputables
Material de oficina	Consumibles X S.A.	30,00
Impresión de documentación	Imprenta X S.A.	120,00
Total		150,00

^{a)} Este capítulo de gastos incluye todos los gastos no contemplados en los conceptos anteriores, por ejemplo: fungible, viajes y dietas, otros,...

COSTES INDIRECTOS DEL PROYECTO

Descripción	Costes	Costes imputables
Gastos comunes	170,00	34,00
Total		34,00

6.- Resumen de costes

Presupuesto Costes Totales	Presupuesto Costes Totales
Personal	18.392
Amortización	330
Subcontratación de tareas	0
Costes de funcionamiento	150
Costes Indirectos	34
Subtotal	18.906
+ IVA (18%)	3.403,07€
Total	22.309,01€

El presupuesto total de este proyecto asciende a la cantidad de **22.309,01 Euros**.

Leganés a 27 de Julio de 2010

Fdo. Adrián López Pérez

Glosario

AJAX (Asynchronous JavaScript And XML): Técnica de desarrollo web para crear aplicaciones enriquecidas con un nivel de interactividad que el alcanzado mediante las peticiones HTTP estándar mediante ejecución de código Javascript en el cliente.

DCU (Diseño Centrado en el Usuario): Proceso de diseño en el que las necesidades y las limitaciones del usuario final de un interfaz se toman en consideración durante todo el proceso de diseño.

DTD (Document Type Definition): En Español "Definición de Tipo de Documento", es una descripción de la estructura de un documento XML (o XHTML como subconjunto del anterior). En el se describen la estructura y el formato de los datos.

JavaBean: Componentes de software reutilizables que se puedan manipular visualmente en una herramienta de construcción. Estos componentes deben de ser clases Java que cumplan las siguientes convenciones: tener un constructor sin argumentos, que sus propiedades sean accesibles mediante métodos `get` y `set` que siguen una convención de nomenclatura estándar y ser serializables.

LRU (Less Recently Used): Algoritmo de reemplazo que selecciona para eliminar los registros que han sido menos accedidos recientemente.

Reflection: Proceso por el cual un programa puede modificar su propia estructura o comportamiento. Para ello el programa accederá al código como si fueran datos. Es una técnica avanzada que se deberá utilizar solamente en el caso de que se tenga buen conocimiento de esta y siempre que el objetivo lo justifique, ya que Reflection hace imposible algunas de las mejoras que se llevan a cabo en la compilación, así como expone código al que no se podría acceder de forma normal, y por lo tanto el sistema podría dejar de funcionar ante una actualización o cambio.

SQL (Structured Query Language): es un lenguaje formal declarativo, estandarizado ISO, para manipular información en una base de datos.

SSL (Secure Sockets Layer): Protocolo criptográfico que proporciona autenticación del servidor y privacidad de los datos transmitidos a través de una conexión a través de Internet.

URL (Uniform Resource Locator): En Español “Localizador uniforme de recursos”, es un formato estandar para nombrar recursos en Internet, posibilitando su localización e identificación. Estos recursos pueden ser tanto textuales, imágenes o elementos multimedia.

Bibliografía

Legislación Alemana. (23 de Julio de 2002). *Decreto sobre Tecnología de la Información Libre de Barreras*. Recuperado el Junio de 2010, de <http://www.einfach-fuer-alle.de/artikel/bitv/>

CTIC. (2010). *t.a.w.* Recuperado el Junio de 2010, de <http://www.tawdis.net>

CTIC. (2010). *t.a.w. monitor*. Recuperado el Junio de 2010, de <http://monitor.fundacionctic.org/tawmonitor/es/index.xhtml>

CTIC. (2010). *Versión descargable de TAW3 (WCAG 1.0)*. Recuperado el Junio de 2010, de <http://www.tawdis.net/tools/accesibilidad/desktop/?lang=es>

AENOR. (Diciembre de 2004). *Norma UNE 139803:2004*. Recuperado el Junio de 2010, de http://www.inteco.es/Accesibilidad/Normativa_1/Descarga/DescargaUNE_139803

ATutor. (2010). *AChecker*. Recuperado el Junio de 2010, de <http://www.atutor.ca/achecker/>

Boletín Oficial del Estado (BOE). (13 de diciembre de 1999). *LEY ORGÁNICA 15/1999, de 13 de diciembre, de Protección de Datos de Carácter Personal*. Recuperado el Junio de 2010, de <http://www.msc.es/organizacion/sns/planCalidadSNS/pdf/transparencia/LOPD19992.pdf>

BOE. (Julio de 2002). *Boletín Oficial del Estado, LEY 34/2002, de 11 de julio, de servicios de la sociedad de la información y de comercio electrónico*. Recuperado el Junio de 2010, de <http://www.boe.es/boe/dias/2002/07/12/pdfs/A25388-25403.pdf>

BOE. (Diciembre de 2003). *Boletín Oficial del Estado, LEY 51/2003, de 2 de diciembre, de igualdad de oportunidades, no discriminación y accesibilidad universal de las personas con discapacidad*. Recuperado el Junio de 2010, de <http://www.boe.es/boe/dias/2003/12/03/pdfs/A43187-43195.pdf>

BOE. (12 de Noviembre de 2007). *REAL DECRETO 1494/2007, de 12 de noviembre, por el que se aprueba el Reglamento sobre las condiciones básicas para el acceso de las personas con discapacidad a las tecnologías, productos y servicios relacionados con la sociedad de la información y medios*. Recuperado el Junio de 2010, de http://www.boe.es/aeboe/consultas/bases_datos/doc.php?coleccion=iberlex&id=2007/19968&txtlen=1000

ESA BSSC. (Mayo de 1996). *Guide to applying the ESA software engineering standards to small software projects*. Recuperado el Junio de 2010, de http://emits.esa.int/emits-doc/e_support/Bssc962.pdf

Eurostat. (1996). *Disability and Social Participation in Europe*.

Fundacion ONCE, Discapnet. (Junio de 2008). *Accesibilidad en los portales Web de los Ayuntamientos de Capitales de Provincia. Revisión Junio de 2008*. Recuperado el Junio de 2010, de http://www.discapnet.es/Castellano/areastematicas/Accesibilidad/Observatorio_infoaccesibilidad/Documents/Tema_13/html/InformeDetalleAyuntamientos2008.htm

Fundación ONCE, Discapnet. (Octubre de 2008). *Accesibilidad en los portales Web de Servicios y Plataformas de eLearning*. Recuperado el Junio de 2010, de http://www.discapnet.es/Castellano/areastematicas/Accesibilidad/Observatorio_infoaccesibilidad/informesInfoaccesibilidad/Paginas/AccesibilidadenlosportalesWebdeServiciosyPlataformasdeLearning.aspx

Fundación ONCE, Discapnet. (Diciembre de 2008). *Accesibilidad Web en los portales de los Ministerios*. Recuperado el Junio de 2010, de http://www.discapnet.es/Castellano/areastematicas/Accesibilidad/Observatorio_infoaccesibilidad/informesInfoaccesibilidad/Paginas/AccesibilidadWebenlosportalesdelosMinisterios.aspx

Fundación ONCE, Discapnet. (2008). *Informes de Infoaccesibilidad*. Recuperado el Junio de 2010, de http://www.discapnet.es/Castellano/areastematicas/Accesibilidad/Observatorio_infoaccesibilidad/informesInfoaccesibilidad/Paginas/default.aspx

Fundación Sidar. (7 de Octubre de 2007). *Legislación Europea sobre Accesibilidad para la Sociedad de la Información*. Recuperado el Junio de 2010, de <http://www.sidar.org/recur/direc/legis/euro.php#alemania>

INTECO. (Febrero de 2008). *Estudio-diagnóstico de accesibilidad y calidad web en la PYME española*. Recuperado el Junio de 2010, de http://www.inteco.es/file/IFrLER_WNidc6KV2OP1O1Q

JBoss Community. (2010). *RichFaces*. Recuperado el Junio de 2010, de <http://www.jboss.org/richfaces>

Moreno López, L. (Febrero de 2010). AWA (Accessibility for Web Applications).

Naciones Unidas. (13 de Diciembre de 2006). *Convención sobre los derechos de las personas con discapacidad*. Recuperado el Mayo de 2010, de <http://www.un.org/spanish/disabilities/default.asp?navid=12&pid=497>

Naciones Unidas. (1948). *Declaración Universal de los Derechos Humanos*. Recuperado el Mayo de 2010, de <http://www.un.org/spanish/Depts/dpi/boletin/humanrights/universaldec.html>

Naciones Unidas. (20 de Diciembre de 1993). *Normas Uniformes sobre la igualdad de oportunidades para las personas con discapacidad*. Recuperado el Mayo de 2010, de <http://www.un.org/esa/socdev/enable/dissreso.htm>

Ocawa. (2010). *Ocawa Web Accessibility Expert*. Recuperado el Junio de 2010, de <http://www.ocawa.com>

Object Management Group. (2009). *Object Management Group - UML*. Recuperado el Mayo de 2010, de Object Management Group - UML: <http://www.uml.org/>

OpenSymphony. (2009). *XWork Documentation*. Recuperado el Junio de 2010, de <http://www.opensymphony.com/xwork/>

Oracle Corporation. (2010). *JavaServer Faces Technology*. Recuperado el Junio de 2010, de <http://java.sun.com/javaee/javaserverfaces/>

Oracle Corporation. (2010). *NetBeans Ide*. Recuperado el Junio de 2010, de <http://www.netbeans.org/>

Red Hat Middleware, LLC. (2010). *Hibernate.org*. Recuperado el Junio de 2010, de <https://www.hibernate.org/>

Red Hat, Inc. (14 de Abril de 2010). *Hibernate Reference Documentation*. Recuperado el Junio de 2010, de http://docs.jboss.org/hibernate/stable/core/reference/en/html_single/

Revilla, O. (10 de Mayo de 2008). *Comparativa de accesibilidad de las WCAG 1.0 y la norma UNE 139803/2004*. Recuperado el Junio de 2010, de <http://itakora.com/comparativa-de-accesibilidad-de-las-wcag-10-y-la-norma-une-1398032004/>

Rodríguez, D. J., Viera, J. G., & Castillo, J. (2007). *Universidad y Accesibilidad. Situación en España*. Recuperado el Junio de 2010, de [http://www.iiisci.org/journal/CV\\$/risci/pdfs/C593AX.pdf](http://www.iiisci.org/journal/CV$/risci/pdfs/C593AX.pdf)

Sidar. (2005). *Hera, Revisando la Accesibilidad con Estilo*. Recuperado el Junio de 2010, de <http://www.sidar.org/hera/>

SpringSource. (2010). *Spring Source Community*. Recuperado el Junio de 2010, de <http://www.springsource.org/>

The Apache Software Foundation. (2009). *Apache Struts*. Recuperado el Junio de 2010, de <http://struts.apache.org/>

The Apache Software Foundation. (2010). *MyFaces Project*. Recuperado el Junio de 2010, de <http://myfaces.apache.org/>

The Eclipse Foundation. (2010). *Eclipse.org home*. Recuperado el Junio de 2010, de <http://www.eclipse.org/>

UE. (13 de marzo de 2001). *Síntesis de la legislación de la UE*. Recuperado el Junio de 2010, de eEurope 2002: http://europa.eu/legislation_summaries/information_society/l24226a_es.htm

UE. (28 de Mayo de 2002). *Síntesis de la legislación de la UE*. Recuperado el Junio de 2010, de eEurope 2005: http://europa.eu/legislation_summaries/information_society/l24226_es.htm

UE. (1 de Junio de 2005). *Síntesis de la legislación de la UE*. Recuperado el Junio de 2010, de i2010: la sociedad de la información y los medios de comunicación al servicio del crecimiento y el empleo: http://europa.eu/legislation_summaries/information_society/c11328_es.htm

UK Statute Laws Database. (1995). *Disability Discrimination Act*. Recuperado el Junio de 2010, de <http://www.statutelaw.gov.uk/content.aspx?activeTextDocId=3330327>

United States Laws. (1998). *Sección 508 United States Laws, Overview of the Rehabilitation Act of 1973. Sections 504 and 508*. Recuperado el Junio de 2010, de <http://www.section508.gov/index.cfm?&FuseAction=Content&ID=12>

W3C. (Septiembre de 2008). *Cascading Style Sheets Level 2 Revision 1 (CSS 2.1) Specification*. Recuperado el Junio de 2010, de <http://www.w3.org/TR/CSS2/>

W3C. (2008). *Conformance Evaluation of Web Sites for Accessibility*. Recuperado el Junio de 2010, de <http://www.w3.org/WAI/eval/conformance.html>

W3C. (Julio de 2010). *Authoring Tool Accessibility Guidelines (ATAG) Overview*. Recuperado el Julio de 2010, de <http://www.w3.org/WAI/intro/atag.php>

W3C. (Octubre de 2009). *Evaluation and Report Language (EARL) Overview*. Recuperado el Junio de 2010, de <http://www.w3.org/WAI/intro/earl.php>

W3C. (Diciembre de 2008). *How to Meet WCAG 2.0*. Recuperado el Junio de 2010, de <http://www.w3.org/WAI/WCAG20/quickref/>

W3C. (Julio de 2005). *User Agent Accessibility Guidelines (UAAG) Overview*. Recuperado el Junio de 2010, de <http://www.w3.org/WAI/intro/uaag.php>

W3C. (Diciembre de 2009). *WAI-ARIA Overview*. Recuperado el Junio de 2010, de <http://www.w3.org/WAI/intro/aria.php>

W3C. (11 de diciembre de 2008). *Web Content Accessibility Guidelines (WCAG) 2.0*. Recuperado el Mayo de 2010, de <http://www.w3.org/TR/WCAG/>

W3C. (Diciembre de 2008). *Web Content Accessibility Guidelines (WCAG) Overview*. Recuperado el Junio de 2010, de <http://www.w3.org/WAI/intro/wcag.php>

W3C. (2010). *Web Accessibilit Initiative*. Recuperado el Mayo de 2010, de <http://www.w3.org/WAI>

W3C. (Enero de 2000). *XHTML 1.0 The Extensible HyperText Markup Language (Second Edition)*. Recuperado el Junio de 2010, de <http://www.w3.org/TR/xhtml1/>

WAI. (5 de Mayo de 1999). *Web Content Accessibility Guidelines 1.0*. Recuperado el Junio de 2010, de <http://www.w3.org/TR/WCAG10/>

Anexo A.

Detalle de las clases

JavaBean

A continuación se detallan los JavaBeans de la aplicación, obviando algunos detalles como la sobrescritura de métodos como toString, compareTo, equals o hashCode:

Tabla A-1: Detalle del JavaBean AutomaticWarningBean

AutomaticWarningBean		
id	Tipo	int
	Descripción	Identificador unívoco de la alerta automática.
	Visibilidad	Privado
site	Tipo	SiteBean
	Descripción	Sitio web monitorizado asociado a esta incidencia, la URL debe pertenecer a este sitio web.
	Visibilidad	Privado
URL	Tipo	String
	Descripción	Página web en la que el usuario ha encontrado el problema.
	Visibilidad	Privado
importance	Tipo	String
	Descripción	Gravedad del problema de accesibilidad, esta clasificación dependerá de las pautas seguidas por módulo de análisis automático de accesibilidad. El módulo integrado por defecto evalúa según las pautas de WCAG 1.0, y por lo tanto la importancia será dada según el grado de prioridad de esta pauta.
	Visibilidad	Privado
checkpoint	Tipo	String
	Descripción	Punto de comprobación de accesibilidad, dependerá de las pautas seguidas por módulo de análisis automático de accesibilidad. El módulo integrado por defecto evalúa según las pautas de WCAG 1.0 y por lo tanto los puntos de comprobación corresponderán a los de WCAG 1.0.
	Visibilidad	Privado
name	Tipo	String

	Descripción	Nombre descriptivo del punto de verificación.
	Visibilidad	Privado
description	Tipo	String
	Descripción	Descripción detallada del punto de verificación, donde se recoge el objetivo de este.
	Visibilidad	Privado
recommendations	Tipo	String
	Descripción	Detalle de las recomendaciones generales para cumplir el punto de verificación, indicando patrones de código o buenas prácticas en función de este.
	Visibilidad	Privado
repair	Tipo	String
	Descripción	Pautas a seguir para reparar los errores específicos encontrados al analizar el punto de verificación.
	Visibilidad	Privado

Tabla A-2: Detalle del JavaBean ClassificationBean

ClassificationBean		
id	Tipo	int
	Descripción	Identificador unívoco de la clasificación.
	Visibilidad	Privado
upperClassification	Tipo	ClassificationBean
	Descripción	Clasificación padre, necesaria para la estructura arbórea.
	Visibilidad	Privado
name	Tipo	String
	Descripción	Nombre descriptivo de la clasificación.
	Visibilidad	Privado
sons	Tipo	Set<ClassificationBean>
	Descripción	Hijos de esta clasificación, facilitan la creación de la estructura arbórea, hibernate es el encargado de resolver estas relaciones sobre la base de datos.
	Visibilidad	Privado

Tabla A-3: Detalle del JavaBean CommentBean

CommentBean		
id	Tipo	int
	Descripción	Identificador unívoco del comentario.
	Visibilidad	Privado
submitDate	Tipo	Date
	Descripción	Fecha y hora en la que el comentario fue enviado a la aplicación.
	Visibilidad	Privado
user	Tipo	UserBean
	Descripción	Usuario que creó el comentario.
	Visibilidad	Privado
textComment	Tipo	String

	Descripción	Texto completo del comentario con todo el contenido de este.
	Visibilidad	Privado

Tabla A-4: Detalle del JavaBean IncidenceBean

IncidenceBean		
id	Tipo	int
	Descripción	Identificador unívoco de la incidencia.
	Visibilidad	Privado
name	Tipo	String
	Descripción	Nombre del usuario que notificó la incidencia.
	Visibilidad	Privado
lastName	Tipo	String
	Descripción	Apellidos del usuario que notificó la incidencia.
	Visibilidad	Privado
email	Tipo	String
	Descripción	Dirección de correo del usuario que notificó la incidencia.
	Visibilidad	Privado
URL	Tipo	String
	Descripción	Página web en la que el usuario ha encontrado el problema.
	Visibilidad	Privado
importance	Tipo	String
	Descripción	Gravedad del problema de accesibilidad, 'Alta' ha sido imposible acceder a la información, 'Media' ha sido difícil acceder a la información o la información no está completa, 'Baja' el acceso a la información podría ser mejorado.
	Visibilidad	Privado
disability	Tipo	String
	Descripción	Discapacidad del usuario que notificó la incidencia.
	Visibilidad	Privado
userAgent	Tipo	String
	Descripción	Agente de usuario que se utilizaba cuando apareció el problema.
	Visibilidad	Privado
resolution	Tipo	String
	Descripción	Resolución que se utilizaba cuando apareció el problema.
	Visibilidad	Privado
description	Tipo	String
	Descripción	Descripción aportada por el usuario afectado por el problema.
	Visibilidad	Privado
repliedUser	Tipo	boolean
	Descripción	Variable que indica si el usuario ha sido respondido o no.
	Visibilidad	Privado
comments	Tipo	Set<CommentBean>
	Descripción	Comentarios del técnico con respecto a la incidencia.
	Visibilidad	Privado
creator	Tipo	UserBean
	Descripción	Usuario que ha creado la incidencia, solo estará establecido en el caso de que esta incidencia haya sido creada por un experto.

	Visibilidad	Privado
site	Tipo	SiteBean
	Descripción	Sitio web monitorizado asociado a esta incidencia, la URL debe pertenecer a este sitio web.
	Visibilidad	Privado
state	Tipo	StateBean
	Descripción	Estado en el que se encuentra la incidencia, esta pasará por un conjunto de estados desde el de nueva creación hasta el de resuelta o descartada, en función de los estados definidos y de la política de estados del equipo de trabajo.
	Visibilidad	Privado

Tabla A-5: Detalle del JavaBean IncidenceStateBean

IncidenceStateBean		
id	Tipo	int
	Descripción	Identificador unívoco del estado de incidencia.
	Visibilidad	Privado
name	Tipo	String
	Descripción	Nombre descriptivo del estado de incidencia.
	Visibilidad	Privado
description	Tipo	String
	Descripción	Descripción del estado, donde se puede especificar su significado, los requisitos de una incidencia para pertenecer a él o cualquier otra información relevante.
	Visibilidad	Privado
incidences	Tipo	Set<IncidenceBean>
	Descripción	Conjunto de las incidencias que se encuentran en este estado.
	Visibilidad	Privado
roles	Tipo	Set<RoleBean>
	Descripción	Conjunto de los roles que tienen acceso a este estado de incidencia.
	Visibilidad	Privado

Tabla A-6: Detalle del JavaBean KnowledgeBean

KnowledgeBean		
id	Tipo	int
	Descripción	Identificador unívoco del artículo de conocimiento.
	Visibilidad	Privado
name	Tipo	String
	Descripción	Nombre descriptivo del artículo de conocimiento.
	Visibilidad	Privado
description	Tipo	String
	Descripción	Contenido del artículo de conocimiento, donde se puede especificar cualquier información necesaria.
	Visibilidad	Privado
classifications	Tipo	Set<KnowledgeClassificationBean>

	Descripción	Conjunto de clasificaciones con las que está relacionado el artículo de conocimiento.
	Visibilidad	Privado

Tabla A-7: Detalle del JavaBean KnowledgeClassificationBean

KnowledgeClassificationBean		
id	Tipo	int
	Descripción	Identificador unívoco de la clasificación del artículo de conocimiento.
	Visibilidad	Privado
name	Tipo	String
	Descripción	Nombre descriptivo de la clasificación.
	Visibilidad	Privado
description	Tipo	String
	Descripción	Texto descriptivo de la clasificación.
	Visibilidad	Privado
father	Tipo	KnowledgeClassificationBean
	Descripción	Clasificación padre, necesaria para la estructura arbórea.
	Visibilidad	Privado
sons	Tipo	Set<KnowledgeClassificationBean>
	Descripción	Hijos de esta clasificación, facilitan la creación de la estructura arbórea, hibernate es el encargado de resolver estas relaciones sobre la base de datos.
	Visibilidad	Privado
knowledge	Tipo	Set<KnowledgeBean>
	Descripción	Instancias de conocimiento que pertenecen a esta clasificación.
	Visibilidad	Privado

Tabla A-8: Detalle del JavaBean RoleBean

RoleBean		
id	Tipo	int
	Descripción	Identificador unívoco del rol de usuario.
	Visibilidad	Privado
name	Tipo	String
	Descripción	Nombre descriptivo del rol de usuario.
	Visibilidad	Privado
description	Tipo	String
	Descripción	Descripción del rol de usuario, donde se puede especificar cualquier información.
	Visibilidad	Privado
users	Tipo	Set<UserBean>
	Descripción	Conjunto de usuarios que pertenecen a este rol.
	Visibilidad	Privado
states	Tipo	Set<IncidenceStateBean>
	Descripción	Conjunto de estados de incidencias que pueden

		ser accedidos mediante este rol.
	Visibilidad	Privado
adminAccess	Tipo	boolean
	Descripción	Identifica si el rol de usuario tiene permisos de administración.
	Visibilidad	Privado
incidencesAccess	Tipo	boolean
	Descripción	Identifica si el rol de usuario tiene permisos de visualización y edición de incidencias.
	Visibilidad	Privado
incidenceCreationAccess	Tipo	boolean
	Descripción	Identifica si el rol de usuario tiene permisos de creación de incidencias.
	Visibilidad	Privado
knowledgeAccess	Tipo	boolean
	Descripción	Identifica si el rol de usuario tiene permisos de gestión de los instancias de conocimiento.
	Visibilidad	Privado
autoWarningAccess	Tipo	boolean
	Descripción	Identifica si el rol de usuario tiene permisos de visualización de las incidencias automáticas.
	Visibilidad	Privado

Tabla A-9: Detalle del JavaBean SiteBean

SiteBean		
id	Tipo	int
	Descripción	Identificador unívoco del sitio web monitorizado.
	Visibilidad	Privado
baseUrl	Tipo	String
	Descripción	Dirección URL a la que pertenecen todas las páginas web que se quieren analizar.
	Visibilidad	Privado
canAccess	Tipo	boolean
	Descripción	Indica si se ha podido acceder al sitio web para su análisis o se ha producido algún error.
	Visibilidad	Privado
warnings	Tipo	Set<AutomaticWarningBean>
	Descripción	Conjunto de alertas de accesibilidad encontradas para este sitio.
	Visibilidad	Privado

Tabla A-10: Detalle del JavaBean UserBean

UserBean		
id	Tipo	int
	Descripción	Identificador unívoco del usuario del sistema.
	Visibilidad	Privado

username	Tipo	String
	Descripción	Nombre de usuario, utilizado para la identificación de este.
	Visibilidad	Privado
password	Tipo	String
	Descripción	Contraseña para este usuario.
	Visibilidad	Privado
roles	Tipo	Set<RoleBean>
	Descripción	Conjunto de roles a los que pertenece este usuario.
	Visibilidad	Privado

Anexo B.

Detalle de los JSP

La estructura utilizada para mostrar los detalles de los ficheros JSP es la siguiente:

Tabla B-1: Leyenda de detalle de JSP

Nombre del JSP	
Descripción de funcionalidad	Explica las acciones que realiza la página JSP y la salida que muestra.
Acciones invocadas	Acciones que se invocan desde esta JSP para incrustar su contenido dentro de la salida.
Enlaces a acciones	Enlaces disponibles dentro de la salida de la JSP a otras acciones, lo que indica la navegación que podrá realizar el usuario. Se excluye por claridad la navegación del menú principal y de las miguitas.

A continuación se detallan los Ficheros JSP utilizados:

Tabla B-2: Detalle del JSP admin.jsp

admin.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un menú desde el que se puede acceder a las diferentes acciones de administración.
Acciones invocadas	menuAction
Enlaces a acciones	adminRoles, adminUsers, adminStates, adminClassification, adminSites

Tabla B-3: Detalle del JSP adminClassification.jsp

adminClassification.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un listado de las clasificaciones de las incidencias, recogiendo su identificador, su nombre y la clasificación padre de esta; un formulario para creación de nuevas clasificaciones y

	controles para la eliminación de las existentes.
Acciones invocadas	menuAction
Enlaces a acciones	addClassification , deleteClassification

Tabla B-4: Detalle del JSP adminRoles.jsp

adminRoles.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un listado de los roles de usuarios, recogiendo su identificador, su nombre y su descripción; un formulario para creación de nuevos roles y controles para la actualización y eliminación de los existentes.
Acciones invocadas	menuAction
Enlaces a acciones	addRoles , updateRoles, deleteRoles

Tabla B-5: Detalle del JSP adminSites.jsp

adminSites.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un listado de los sitios monitorizados, recogiendo su identificador, su dirección URL base y su estado (Correcto o con errores); un formulario para creación de nuevos sitios y controles para la eliminación de los existentes.
Acciones invocadas	menuAction
Enlaces a acciones	addSites , deleteSites

Tabla B-6: Detalle del JSP adminStates.jsp

adminStates.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un listado de los estados de las incidencias, recogiendo su identificador, su nombre y su descripción un formulario para creación de nuevos estados y controles para modificar los roles que tienen acceso y eliminar los existentes.
Acciones invocadas	menuAction
Enlaces a acciones	addStates , deleteRoles, showRolesOfStates

Tabla B-7: Detalle del JSP adminUsers.jsp

adminUsers.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un listado de los roles de usuarios, recogiendo su identificador,

	su nombre de usuario y sus roles; un formulario para creación de nuevos usuarios y controles para la actualización y eliminación de los existentes.
Acciones invocadas	menuAction
Enlaces a acciones	addUsers , updateUsers, deleteUsers

Tabla B-8: Detalle del JSP createExpertIncidence.jsp

createExpertIncidence.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un formulario con los campos necesarios para la creación de una nueva incidencia por parte de un experto.
Acciones invocadas	menuAction
Enlaces a acciones	createNewExpertIncidence

Tabla B-9: Detalle del JSP createIncidence.jsp

createIncidence.jsp	
Descripción de funcionalidad	Genera la estructura básica de la web para usuarios externos, carente de menú de navegación. En la sección principal de la página muestra un formulario con los campos necesarios para la creación de una nueva incidencia por parte de un usuario del sitio web.
Acciones invocadas	Ninguna
Enlaces a acciones	createNewIncidence

Tabla B-10: Detalle del JSP createKnowledge.jsp

createKnowledge.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un formulario con los campos necesarios para la creación de un nuevo artículo de conocimiento o, en el caso de que estos campos ya hayan sido definidos, muestra el árbol utilizado para definir su clasificación. Permite asimismo realizar las llamadas y mostrar los cambios oportunos para el manejo del árbol de clasificaciones.
Acciones invocadas	menuAction
Enlaces a acciones	createKnowledge, editKnowledge, createKnowledgeClassification, showKnowClassif

Tabla B-11: Detalle del JSP login.jsp

login.jsp

Descripción de funcionalidad	Genera una estructura reducida de la web, carente de menú de navegación y con una pequeña ventana para el formulario de <i>login</i> , donde se pueden introducir las credenciales de acceso
Acciones invocadas	Ninguna
Enlaces a acciones	login

Tabla B-12: Detalle del JSP menu.jsp

menu.jsp	
Descripción de funcionalidad	Genera la estructura del menú principal de navegación en función de los datos obtenidos de la acción, que dependerán de los permisos del usuario.
Acciones invocadas	Ninguna
Enlaces a acciones	start, listIncidentes, inputNewExpertIncidence, showSites, showKnowClassif, admin, logout Estas acciones con las mostradas a un usuario con todos los permisos, se podrán mostrar menos en función de los permisos específicos del usuario.

Tabla B-13: Detalle del JSP myIncidentes.jsp

myIncidentes.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un listado de las incidencias para las que el usuario tiene permiso de visualización, clasificadas por estado y por sitio monitorizado, recogiendo su identificador, su descripción, su clasificación y su importancia. Se podrá acceder a la visualización y edición de las incidencias.
Acciones invocadas	menuAction
Enlaces a acciones	showIncidence

Tabla B-14: Detalle del JSP showCheckpoint.jsp

showCheckpoint.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un listado de los puntos de verificación que contienen alertas automáticas para el sitio web monitorizado seleccionado, recogiendo su nombre, su importancia y sus recomendaciones. Se podrá acceder a las alertas pertenecientes a los puntos de verificación mostrados.
Acciones invocadas	menuAction
Enlaces a acciones	showWarnings

Tabla B-15: Detalle del JSP showClassifKnowledge.jsp

showClassifKnowledge.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra el árbol de clasificaciones de instancias de conocimiento y un botón para el acceso a la creación de nuevo conocimiento. Permite asimismo realizar las llamadas y mostrar los cambios oportunos para el manejo del árbol de clasificaciones. El árbol de clasificaciones permite acceder a los listados de artículos para esas clasificaciones.
Acciones invocadas	menuAction
Enlaces a acciones	showKnowledge, showKnowClassif, createKnowledge

Tabla B-16: Detalle del JSP showIncidence.jsp

showIncidence.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un formulario con los campos necesarios para la edición de una incidencia, presentando los datos anteriores de esta.
Acciones invocadas	menuAction
Enlaces a acciones	editIncidence

Tabla B-17: Detalle del JSP showKnowledge.jsp

showKnowledge.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un listado de los instancias de conocimiento de la clasificación seleccionada, detallando el nombre y una parte del contenido. Se podrá acceder al contenido completo de estas instancias de conocimiento.
Acciones invocadas	menuAction
Enlaces a acciones	showSpecificKnowledge

Tabla B-18: Detalle del JSP showSites.jsp

showSites.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un listado de los sitios web monitorizados, detallando la URL base del sitio web y su estado (Correcto o con errores). Se podrá acceder a visualizar los puntos de verificación con alertas dentro de esos sitios.
Acciones invocadas	menuAction
Enlaces a acciones	showCheckpoints

Tabla B-19: Detalle del JSP showSpecificKnowledge.jsp

showSpecificKnowledge.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra los campos del artículo de conocimiento seleccionado. No permite editar sus valores.
Acciones invocadas	menuAction
Enlaces a acciones	Ninguna

Tabla B-20: Detalle del JSP showWarn.jsp

showWarn.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra los campos de la alerta de accesibilidad automática seleccionada. No permite editar sus valores.
Acciones invocadas	menuAction
Enlaces a acciones	Ninguna

Tabla B-21: Detalle del JSP showWarns.jsp

showWarns.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un listado de alertas automáticas de accesibilidad del punto de verificación seleccionado, detallando la dirección URL donde se han encontrado y el nombre de la alerta. Se podrá acceder a las alertas específicas.
Acciones invocadas	menuAction
Enlaces a acciones	showWarning

Tabla B-22: Detalle del JSP start.jsp

start.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra información sobre la aplicación.
Acciones invocadas	menuAction
Enlaces a acciones	Ninguna

Tabla B-23: Detalle del JSP successCommit.jsp

successCommit.jsp	
Descripción de funcionalidad	Genera la estructura básica de la web para usuarios externos, carente de menú de navegación. En la sección principal de la página muestra un mensaje indicando que la incidencia ha sido enviada correctamente.
Acciones invocadas	Ninguna
Enlaces a acciones	Ninguna

Tabla B-24: Detalle del JSP successCommitExpert.jsp

successCommitExpert.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un mensaje indicando que la incidencia ha sido enviada correctamente.
Acciones invocadas	menuAction
Enlaces a acciones	Ninguna

Tabla B-25: Detalle del JSP updateRoles.jsp

updateRoles.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un formulario con los campos del rol que se desea editar.
Acciones invocadas	menuAction
Enlaces a acciones	updateRoles

Tabla B-26: Detalle del JSP updateStates.jsp

updateStates.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestran controles para vincular o desvincular los roles de usuario que tienen permiso para acceso a las incidencias de este estado de incidencia en particular.
Acciones invocadas	menuAction
Enlaces a acciones	addRoleToStates, deleteRoleToStates

Tabla B-27: Detalle del JSP updateUsers.jsp

updateUsers.jsp	
Descripción de funcionalidad	Genera la estructura básica de web interna, con el menú principal. En la sección principal de la página muestra un formulario con los campos del usuario que se desea editar.
Acciones invocadas	menuAction
Enlaces a acciones	updateUsers

Anexo C.

Detalle de las clases de acción

La plantilla seguida para la especificación de las clases de acción es la siguiente:

Tabla C-1: Leyenda de detalle de la clase de acción.

Nombre de la clase	
Descripción de funcionalidad	Descripción del objetivo y funcionalidad genérica que realiza esta clase.
Métodos de acción	Detalle de los métodos públicos de la clase destinados a recibir las llamadas de las acciones Struts, indicando los datos más relevantes de estos. Estos métodos siempre devuelven un objeto de tipo String y no admiten parámetros.
Métodos de acceso a datos	Detalle de los métodos públicos de la clase destinados al manejo de datos, indicando la información más relevantes de estos. Dichos métodos serán accedidos desde la vista a través de <i>tags</i> de Struts invocaciones de JSP y serán del tipo <i>get</i> y <i>set</i> siguiendo las normas explicadas en 4.4.4 Identificación de subsistemas.
Métodos auxiliares	Detalle de los métodos privados utilizados dentro de la clase para realizar operaciones complejas.
Acciones que la invocan	Listado de las acciones de Struts que invocan a esta clase.
JSP resultantes	Listado de los JSP que se utilizan como componentes de la vista para generar la salida de esta clase.

A continuación se detallan las clases de acción:

Tabla C-2: Detalle de la clase de acción *AdminClassification*

AdminClassification		
Descripción de funcionalidad	Clase encargada de gestionar las operaciones de administración de clasificaciones de incidencias. Presenta métodos que realizan las acciones de mostrar, añadir y eliminar clasificaciones.	
Métodos de acción	admin	Obtiene un listado de las clasificaciones de las incidencias

		del sistema y lo almacena en el campo <code>classificationList</code> .
	add	Crea una nueva clasificación, haciendo uso de los campos <code>classificationName</code> para el nombre de esta y <code>classificationUpper</code> para su clasificación superior. Si alguno de los campos no existe no realiza ninguna acción. Por último obtiene un listado de las clasificaciones del sistema y lo almacena en el campo <code>classificationList</code> .
	delete	Elimina una clasificación, haciendo uso del campo <code>classificationToDelete</code> para obtener el identificador de la clasificación que se desea eliminar. Si el campo no existe no realiza ninguna acción. Por último obtiene un listado de las clasificaciones del sistema y lo almacena en el campo <code>classificationList</code> .
Métodos de acceso a datos	<code>getClassificationList() : List<ClassificationBean></code>	
	Obtiene la lista de las clasificaciones del sistema.	
	<code>setClassificationName(String)</code>	
	Establece el nombre para la nueva clasificación.	
	<code>setClassificationToDelete(String)</code>	
	Establece el identificador de la clasificación que se va a eliminar.	
	<code>setClassificationUpper(String)</code>	
	Establece el identificador de la clasificación superior a la nueva que se desea crear.	
Métodos auxiliares	Ninguno	
Acciones que la invocan	<code>adminClassification,</code> <code>addClassification,</code> <code>deleteClassification</code>	
JSP resultantes	<code>adminClassification.jsp</code>	

Tabla C-3: Detalle de la clase de acción `AdminRoles`

AdminRoles		
Descripción de funcionalidad	Clase encargada de gestionar las operaciones de administración de los roles del sistema. Presenta métodos que realizan las acciones de mostrar, añadir, actualizar y eliminar los roles.	
Métodos de acción	admin	Obtiene un listado de los roles del sistema y lo almacena en el campo <code>roleList</code> .
	add	Crea un nuevo rol, haciendo uso de los campos <code>roleName</code> para el nombre de este y <code>roleDescription</code> para su descripción, así como los campos <code>adminAccessToUpdate,</code> <code>autoWarningAccessToUpdate,</code> <code>knowledgeAccessToUpdate,</code> <code>incidencesAccessToUpdate</code> e <code>incidenceCreationAccessToUpdate</code> para establecer los permisos. Si alguno de los campos no existe no realiza ninguna acción. Por último obtiene un listado de los roles del sistema y lo almacena en el campo <code>roleList</code> .
	update	Obtiene los datos de un rol o modifica un rol en función del campo <code>updating,</code> haciendo uso del campo <code>roleIdToUpdate</code> para identificar el rol que se desea

		modificar, los campos <code>roleNameToUpdate</code> y <code>roleDescriptionToUpdate</code> son utilizados para obtener o establecer los nuevos valores, así como los campos <code>adminAccessToUpdate</code> , <code>autoWarningAccessToUpdate</code> , <code>knowledgeAccessToUpdate</code> , <code>incidencesAccessToUpdate</code> e <code>incidenceCreationAccessToUpdate</code> para obtener o establecer los nuevos permisos. Por último obtiene un listado de los roles del sistema y lo almacena en el campo <code>roleList</code> .
	delete	Elimina un rol, haciendo uso del campo <code>roleToDelete</code> para obtener el identificador del rol que se desea eliminar. Si el campo no existe no realiza ninguna acción. Por último obtiene un listado de las clasificaciones del sistema y lo almacena en el campo <code>roleList</code> .
Métodos de acceso a datos	<code>getRoleDescriptionToUpdate() : String</code>	
	Obtiene el valor anterior del campo de descripción del rol.	
	<code>getRoleIdToUpdate() : String</code>	
	Obtiene el identificador del rol.	
	<code>getRoleList() : List<RoleBean></code>	
	Obtiene un listado de los roles del sistema.	
	<code>getRoleName() : String</code>	
	Utilizado para obtener el nombre anterior del rol, agilizando así la actualización.	
	<code>getRoleNameToUpdate() : String</code>	
	Obtiene el valor anterior del campo de nombre del rol.	
	<code>isAdminAccessToUpdate() : boolean</code>	
	Obtiene el valor anterior de permiso de administración del rol.	
	<code>isAdminAutoWarningsAccessToUpdate() : boolean</code>	
	Obtiene el valor anterior de permiso de acceso a las alertas de accesibilidad del rol.	
	<code>isIncidenceCreationAccessToUpdate() : boolean</code>	
	Obtiene el valor anterior de permiso de creación de incidencias del rol.	
	<code>isIncidencesAccessToUpdate() : boolean</code>	
	Obtiene el valor anterior de permiso de acceso a las incidencias del rol.	
	<code>isKnowledgeAccessToUpdate() : boolean</code>	
	Obtiene el valor anterior de permiso de gestión del conocimiento del rol.	
	<code>setAdminAccessToUpdate(boolean)</code>	
	Establece el nuevo valor para el permiso de administración del rol.	
	<code>setAutoWarningAccessToUpdate(boolean)</code>	
	Establece el nuevo valor para el permiso de acceso a las alertas automáticas del rol.	
	<code>setDescriptionToUpdate(String)</code>	
	Establece el nuevo valor para el campo de descripción del rol.	
	<code>setIncidenceCreationAccessToUpdate(boolean)</code>	

	Establece el nuevo valor para el permiso de creación de incidencias del rol.
	<code>setIncidencesAccessToUpdate(boolean)</code>
	Establece el nuevo valor para el permiso de acceso a las incidencias del rol.
	<code>setKnowledgeAccessToUpdate(boolean)</code>
	Establece el nuevo valor para el permiso de gestión del conocimiento del rol.
	<code>setRoleDescription(String)</code>
	Establece el valor del campo descripción para el nuevo rol.
	<code>setRoleDescriptionToUpdate(String)</code>
	Establece el nuevo valor para el campo nombre del rol.
	<code>setRoleIdToUpdate(String)</code>
	Establece el identificador del rol que se desea editar.
	<code>setRoleName(String)</code>
	Establece el valor del campo nombre para el nuevo rol.
	<code>setRoleNameToUpdate(String)</code>
	Establece el nuevo valor para el campo nombre del rol.
	<code>setRoleToDelete(String)</code>
	Establece el identificador del rol que se desea eliminar.
	<code>setUpdating(String)</code>
	Se establece un valor para indicar que se están enviando los datos para la modificación.
Métodos auxiliares	Ninguno
Acciones que la invocan	<code>adminRoles</code> , <code>addRoles</code> , <code>updateRoles</code> , <code>deleteRoles</code>
JSP resultantes	<code>adminRoles.jsp</code> , <code>updateRoles.jsp</code>

Tabla C-4: Detalle de la clase de acción `AdminSites`

AdminSites		
Descripción de funcionalidad	Clase encargada de gestionar las operaciones de administración de los sitios monitorizados del sistema. Presenta métodos que realizan las acciones de mostrar, añadir y eliminar los sitios.	
Métodos de acción	admin	Obtiene un listado de los sitios monitorizados del sistema y lo almacena en el campo <code>siteList</code> .
	add	Crea un nuevo sitio monitorizado, haciendo uso del campo <code>baseUrl</code> para la dirección URL base de este. Si el campo no existe o existe otro sitio con la misma dirección base, no realiza ninguna acción. Por último obtiene un listado de los roles del sistema y lo almacena en el campo <code>siteList</code> .
	delete	Elimina el sitio monitorizado especificado en el campo <code>idToDelete</code> para obtener el identificador de este. Si el campo no existe no realiza ninguna acción. Por último obtiene un listado de los roles del sistema y lo almacena en el campo <code>siteList</code> .
	<code>getSiteList() : List<SiteBean></code>	
	Obtiene un listado de los sitios monitorizados del sistema.	
	<code>setBaseUrl(String)</code>	

Métodos auxiliares	Establece el valor de la dirección URL base para el nuevo sitio.
	<code>setIdToDelete(String)</code>
	Establece el identificador del sitio que se desea eliminar.
	Ninguno
Acciones que la invocan	<code>adminSites, addSites, deleteSites</code>
JSP resultantes	<code>adminSites.jsp</code>

Tabla C-5: Detalle de la clase de acción `AdminStates`

AdminStates		
Descripción de funcionalidad	Clase encargada de gestionar las operaciones de administración de los estados de las incidencias del sistema. Presenta métodos que realizan las acciones de mostrar, añadir y eliminar los estados, así como añadir y eliminar los roles asociados a estos.	
Métodos de acción	admin	Obtiene un listado de los roles del sistema y lo almacena en el campo <code>stateList</code> .
	add	Crea un nuevo estado, haciendo uso de los campos <code>stateName</code> para el nombre de este y <code>stateDescription</code> para su descripción. Si el campo <code>stateName</code> no existe o ya existe otro con el mismo valor de devuelve un error. Por último obtiene un listado de los estados del sistema y lo almacena en el campo <code>stateList</code> .
	delete	Elimina un estado, haciendo uso del campo <code>stateToDelete</code> para obtener el identificador del estado que se desea eliminar. Si el campo no existe no realiza ninguna acción. Por último obtiene un listado de los estados del sistema y lo almacena en el campo <code>stateList</code> .
	showRolesOf	Obtiene un listado de los roles que tienen permiso para visualización de incidencias y lo almacena en <code>roleList</code> , almacenando en <code>isAssignedList</code> para cada uno de ellos si está asignado al estado identificado por <code>stateToEdit</code> .
	addRoleTo	Asocia el rol con el identificador <code>roleToUse</code> con el estado con el identificado <code>stateToEdit</code> . Si alguno de los campos no existe no realiza ninguna acción.
	deleteRoleTo	Desasocia el rol con el identificador <code>roleToUse</code> con el estado con el identificado <code>stateToEdit</code> . Si alguno de los campos no existe no realiza ninguna acción.
Métodos de acceso a datos	<code>getIsAssignedList() : List<Boolean></code>	
	Obtiene la lista que identifica los roles asociados al estado.	
	<code>getRoleList() : List<RoleBean></code>	
	Obtiene un listado de los roles del sistema que tienen permisos para visualización de incidencias.	
	<code>getStateList() : List<RoleBean></code>	
	Obtiene un listado de los estados del sistema.	
	<code>getStateName() : String</code>	

	Obtiene el nombre del estado del sistema.
	<code>getStateToEdit() : String</code>
	Obtiene el identificador del estado a editar.
	<code>setStateToEdit(String)</code>
	Especifica el identificador del estado a editar.
	<code>setRoleToUse(String)</code>
	Establece el rol que se desea asociar o desasociar al estado.
	<code>setStateDescription(String)</code>
	Establece el valor para el campo de descripción del nuevo estado.
	<code>setStateName(String)</code>
	Establece el valor para el campo de nombre del nuevo estado.
	<code>setStateToDelete(String)</code>
	Establece el identificador del estado que se desea eliminar.
	<code>setStateToEdit(String)</code>
	Establece el identificador del estado que se desea editar.
Métodos auxiliares	Ninguno
Acciones que la invocan	<code>adminStates,</code> <code>addStates,</code> <code>deleteStates,</code> <code>showRolesOfStates,</code> <code>addRoleToStates,</code> <code>deleteRoleToStates</code>
JSP resultantes	<code>adminStates.jsp,</code> <code>updateStates.jsp</code>

Tabla C-6: Detalle de la clase de acción `AdminUsers`

AdminUsers		
Descripción de funcionalidad	Clase encargada de gestionar las operaciones de administración de los usuarios del sistema. Presenta métodos que realizan las acciones de mostrar, añadir, actualizar y eliminar los usuarios.	
Métodos de acción	admin	Obtiene un listado de los usuarios del sistema y lo almacena en el campo <code>userList</code> .
	add	Crea un nuevo usuario, haciendo uso de los campos <code>username</code> para el nombre de este y <code>password</code> para su contraseña. Si alguno de los campos no existe devuelve un error. Por último obtiene un listado de los usuarios del sistema y lo almacena en el campo <code>userList</code> .
	update	Obtiene los datos de un usuario o modifica este en función del campo <code>updating</code> , haciendo uso del campo <code>userIdToUpdate</code> para identificar el usuario que se desea modificar y los campos <code>userNameToUpdate</code> y <code>userPasswordToUpdate</code> , que son utilizados para obtener o establecer los nuevos valores, así como los campos <code>selectedRoleMap</code> , utilizando para mostrar los roles asociados a este usuario y <code>newSelectedRoles</code> , establecer los nuevos roles asociados. Por último obtiene un listado de los usuarios del sistema y lo almacena en el campo <code>userList</code> .
	delete	Elimina un usuario, haciendo uso del campo <code>userToDelete</code> para obtener el identificador del usuario que se desea eliminar. Si el campo no existe no realiza ninguna acción. Por último obtiene un listado de los usuarios del sistema y lo almacena en el campo <code>userList</code> .

Métodos de acceso a datos	getSelectedRoleMap() : Map<RoleBean, Boolean>	
	Obtiene el conjunto de los roles del sistema, junto con el booleano que indica la asociación actual de este usuario con cada rol.	
	getUserIdToUpdate() : String	
	Obtiene el identificador del usuario a editar.	
	getUserList() : List<UserBean>	
	Obtiene el listado de los usuarios del sistema.	
	getUserNameToUpdate() : String	
	Obtiene el valor anterior del campo de nombre del usuario.	
	getUserPasswordToUpdate() : String	
	Obtiene el valor anterior del campo de contraseña del usuario.	
	setNewSelectedRoles(String[])	
	Establece los roles asociados al usuario como una matriz de cadenas de caracteres.	
	setPassword(String)	
	Establece el valor para la contraseña para el nuevo usuario.	
	setUpdating(String)	
	Se establece un valor para indicar que se están enviando los datos para la modificación.	
	setUserIdToUpdate(String)	
	Establece el identificador del usuario que se desea editar.	
	setUsername(String)	
	Establece el valor del campo nombre para el nuevo usuario.	
	setUserNameToUpdate(String)	
	Establece el nuevo valor para el campo nombre del usuario.	
	setUserPasswordToUpdate(String)	
	Establece el nuevo valor para el campo contraseña del usuario.	
Métodos auxiliares	setUserToDelete(String)	
	Establece el identificador del usuario que se desea eliminar.	
Acciones que la invocan	fillUserList()	Obtiene un listado de los usuarios del sistema y lo almacena en el campo userList.
	adminUsers, addUsers, updateUsers, deleteUsers	
JSP resultantes	adminUsers.jsp, updateUsers.jsp	

Tabla C-7: Detalle de la clase de acción **IncidenceAction**

IncidenceAction		
Descripción de funcionalidad	Clase encargada de la obtención de los datos necesarios para mostrar los datos de las incidencias del sistema.	
Métodos de acción	execute	Obtiene los datos de la incidencia identificada por el parámetro <code>incidenceId</code> y lo almacena en el campo <code>incidence</code> . Almacena un listado de las clasificaciones de incidencias en <code>classifications</code> , un listado de los estados del sistema en <code>stateList</code> y el estado actual de la incidencia en <code>assignedState</code> .
Métodos de acceso a datos	getAsignedState() : IncidenceState	
	Obtiene el estado de la incidencia consultada.	

	<code>getClassifications() : List<String></code>
	Obtiene el listado de las clasificaciones de incidencias.
	<code>getIncidence() : IncidenceBean</code>
	Obtiene la incidencia consultada.
	<code>getStateList() : List<IncidenceStateBean></code>
	Obtiene el listado de los estados de las incidencias del sistema.
Métodos auxiliares	Ninguno
Acciones que la invocan	<code>showIncidence</code>
JSP resultantes	<code>showIncidence.jsp</code>

Tabla C-8: Detalle de la clase de acción `IncidenceBean`

IncidenceBean		
Descripción de funcionalidad	Clase encargada además de manejo de datos como <code>JavaBean</code> , de gestionar la creación de nuevas incidencias y de edición de los datos de las existentes.	
Métodos de acción	create	Crea una nueva incidencia en base a los campos <code>url</code> , <code>description</code> , <code>disability</code> , <code>name</code> , <code>lastname</code> , <code>email</code> , <code>userAgent</code> , <code>resolution</code> , <code>importante</code> y <code>classificationString</code> . Si los campos <code>url</code> o <code>description</code> están vacíos se genera un error.
	edit	Modifica una incidencia existente en base a los campos <code>url</code> , <code>description</code> , <code>disability</code> , <code>name</code> , <code>lastname</code> , <code>email</code> , <code>userAgent</code> , <code>resolution</code> , <code>importante</code> , <code>classificationString</code> , <code>newComment</code> y <code>newAssignedState</code> . Si los campos <code>url</code> o <code>description</code> están vacíos se genera un error. Si el campo <code>delete</code> es <code>true</code> la incidencia es eliminada.
Métodos de acceso a datos	<code>getClassification() : ClassificationBean</code>	
	Obtiene la clasificación actual de la incidencia consultada.	
	<code>getComments() : Set<CommentBean></code>	
	Obtiene el conjunto de los comentarios para esta incidencia.	
	<code>getCreator() : UserBean</code>	
	Obtiene el usuario que ha creado la incidencia, en el caso de que esta haya sido creada por un usuario experto en accesibilidad y no mediante el acceso público.	
	<code>getDescription() : String</code>	
	Obtiene el campo <code>description</code> .	
	<code>getDissability() : String</code>	
	Obtiene el campo <code>dissability</code> .	
	<code>getEmail() : String</code>	
	Obtiene el campo <code>email</code> .	
	<code>getId() : String</code>	
	Obtiene el identificador único de la incidencia.	
	<code>getImportance() : String</code>	
	Obtiene el campo <code>importance</code> .	
	<code>getLastName() : String</code>	
	Obtiene el campo <code>lastName</code> .	

	getName() : String
	Obtiene el campo <code>name</code> .
	getResolution() : String
	Obtiene el campo <code>resolution</code> .
	getSite() : SiteBean
	Obtiene el sitio web monitorizado al que pertenece la incidencia.
	getState() : IncidenceStateBean
	Obtiene el estado en el que se encuentra la incidencia.
	getUrl() : String
	Obtiene la dirección URL especificada en la incidencia.
	getUserAgent() : String
	Obtiene el campo de agente de usuario o navegador especificado.
	setClassificationString(String)
	Establece la clasificación actual de la incidencia consultada.
	setDelete(boolean)
	Establece el campo <code>delete</code> , el cual indica si la incidencia debe ser eliminada.
	setDescription(String)
	Establece el campo <code>description</code> .
	setDissability(String)
	Establece el campo <code>dissability</code> .
	setEmail(String)
	Establece el campo <code>email</code> .
	setId(String)
	Indica el identificador de la incidencia que se desea modificar o eliminar.
	setImportance(String)
	Establece el campo <code>importance</code> .
	setLastName(String)
	Establece el campo <code>lastName</code> .
	setName(String)
	Establece el campo <code>name</code> .
	setNewAssignedState(String)
	Establece el estado al que la incidencia se debe asignar.
	setNewComment(String)
	Establece el texto del nuevo comentario que se debe incluir.
	setResolution(String)
	Establece el campo <code>resolution</code> .
	getUrl() : SiteBean
	Establece la dirección URL especificada en la incidencia.
	setUserAgent(String)
	Establece el campo de agente de usuario o navegador especificado.
Métodos auxiliares	Ninguno
Acciones que la invocan	showIncidence
JSP resultantes	showIncidence.jsp

Tabla C-9: Detalle de la clase de acción KnowledgeAction

KnowledgeAction		
Descripción de funcionalidad	Clase encargada de gestionar las instancias de conocimiento, su creación, la navegación por las categorías de estos, su recuperación, etc.	
Métodos de acción	create	Crea un nuevo artículo de contenido en base a los campos <code>name</code> y <code>description</code> , <code>disability</code> , <code>name</code> , <code>lastname</code> , <code>email</code> , <code>userAgent</code> , <code>resolution</code> , <code>importante</code> y <code>classificationString</code> . Si el campo <code>name</code> está vacío se genera un error.
	createClassification	Para una instancia indicado por <code>id</code> , se asignará a la clasificación de contenido especificada por <code>classifId</code> .
	getClassifications	Obtiene un árbol de objetos de clasificaciones de conocimiento y lo almacena en <code>knowledgeRoot</code> , este será procesado en la vista para generar el árbol gráfico.
	getKnowledge	Obtiene el listado de las instancias de conocimiento para la categoría identificada por <code>classifId</code> y lo almacena en <code>knowledgeList</code> .
	getSpecificKnowledge	Obtiene el artículo de conocimiento identificado por <code>knowledgeId</code> y lo almacena en el campo <code>knowledgeItem</code> .
Métodos de acceso a datos	<code>getClassification() : KnowledgeClassificationBean</code>	
	Obtiene la clasificación actual del artículo de conocimiento.	
	<code>getClassifId() : String</code>	
	Obtiene el identificador de la clasificación seleccionada.	
	<code>getKnowledgeItem() : KnowledgeBean</code>	
	Obtiene el artículo de conocimiento almacenado en <code>knowledgeItem</code> .	
	<code>getKnowledgeList() : List<KnowledgeBean></code>	
	Obtiene el listado de instancias de conocimiento almacenado en <code>knowledgeBean</code> .	
	<code>getKnowledgeRoot() : KnowledgeClassificationBean</code>	
	Obtiene el árbol de clasificación de conocimiento almacenado en <code>knowledgeRoot</code> .	
	<code>getTokens() : Set<String></code>	
	Obtiene el conjunto de cadenas de caracteres necesarias para el árbol.	
	<code>getTreeState() : String</code>	
	Obtiene la cadena que identifica el estado de despliegue del árbol.	
	<code>setClassifId(String)</code>	
	Establece la clasificación seleccionada en el árbol.	
	<code>setDescription(String)</code>	
	Establece el campo <code>description</code> del nuevo artículo de contenido.	
	<code>setExpandId(String)</code>	
	Establece el nivel del árbol que se desea desplegar.	
	<code>setId(String)</code>	
	Establece el identificador del artículo al que se le va a establecer la clasificación.	
	<code>setName(String)</code>	
	Establece el campo <code>name</code> del nuevo artículo de contenido.	

Métodos auxiliares	setTreeState(String)	
	Establece la cadena que identifica el estado de despliegue del árbol.	
	setUnExpandId(String)	
	Establece el nivel del árbol que se desea replegar.	
	loadClassifications	Genera las cadenas de caracteres almacenadas en tokens de manera recursiva a partir del árbol de clasificaciones.
Acciones que la invocan	showKnowClassif, showKnowledge, showSpecificKnowledge, editKnowledge, newKnowledge, createKnowledge, createKnowledgeClassification	
JSP resultantes	showClassifKnowledge.jsp, showKnowledge.jsp, showSpecificKnowledge.jsp, createKnowledge.jsp	

Tabla C-10: Detalle de la clase de acción ListIncidentes

ListIncidentes		
Descripción de funcionalidad	Clase encargada de proporcionar un listado de incidencias en función de los permisos del usuario que lo solicite.	
Métodos de acción	execute	Obtiene el listado de incidencias para el usuario en base a los permisos de que disponga este, almacenando el resultado en el campo incidences. Las incidencias que almacenan en una estructura que contiene: los sitios web monitorizados para los que existen incidencias y las incidencias de estos clasificadas por estado, mostrando solo los que el usuario tiene acceso.
Métodos de acceso a datos	getIncidentes() : Map<IncidenceStateBean, Set<Map<String, Set<IncidenceBean>>>>	
	Obtiene la estructura que contiene los sitios web monitorizados para los que existen incidencias y las incidencias de estos clasificadas por estado.	
Métodos auxiliares	Ninguno	
Acciones que la invocan	listIncidentes	
JSP resultantes	myIncidences.jsp	

Tabla C-11: Detalle de la clase de acción Login

Login		
Descripción de funcionalidad	Clase encargada de validación de los datos de acceso introducidos en la interfaz de acceso.	
Métodos de acción	execute	Comprueba que los datos introducidos en los campos usernameLogin y passwordLogin no sean nulos.
	validate	Valida los datos introducidos en el formulario de acceso, comprobando que estos no sean vacíos y se encuentren en la base de datos. En caso de ser correctos vincula la sesión HttpSession con el usuario, permitiendo al usuario permanecer autenticado en la aplicación durante toda su navegación. El objeto HttpSession de Java permite adaptar la forma en que se mantiene la sesión a las capacidades o preferencias del servidor, de esta forma se podrá usar cookies o reescritura de la dirección URL.

Métodos de acceso a datos	setPasswordLogin(String)
	Establece la contraseña utilizada para el acceso al sistema.
Métodos auxiliares	setUsernameLogin(String)
	Establece el nombre de usuario utilizado para el acceso al sistema.
Acciones que la invocan	Ninguno
JSP resultantes	login.jsp

Tabla C-12: Detalle de la clase de acción Logout

Logout		
Descripción de funcionalidad	Clase encargada de procesar la petición de salida del usuario del sistema.	
Métodos de acción	execute	Obtiene la sesión del usuario por medio del objeto <code>HttpSession</code> y la desvincula del sistema.
Métodos de acceso a datos	Ninguno	
Métodos auxiliares	Ninguno	
Acciones que la invocan	logout	
JSP resultantes	Ninguna, redirección a la acción "login"	

Tabla C-13: Detalle de la clase de acción MenuInterface

MenuInterface		
Descripción de funcionalidad	Clase encargada de la creación del menú principal de la aplicación en función de los permisos del usuario que acceda.	
Métodos de acción	execute	Obtiene el usuario actual y sus permisos, incluyendo los enlaces genéricos, así como un conjunto de enlaces de menú para cada uno de estos permisos, almacenándolos en el campo <code>links</code> . Asimismo obtiene el nombre del usuario actual y lo almacena en el campo <code>username</code> , para posterior consulta desde la vista.
Métodos de acceso a datos	getLinks () : List<String>	
	Obtiene una lista de los enlaces almacenados en el campo <code>links</code> .	
	getUsername () : String	
	Obtiene el nombre del usuario actual almacenado en el campo <code>username</code> .	
Métodos auxiliares	Ninguno	
Acciones que la invocan	menuAction	
JSP resultantes	menu.jsp	

Tabla C-14: Detalle de la clase de acción StartInterface

StartInterface		
Descripción de	Clase encargada de la creación de la interfaz de entrada a la	

funcionalidad	aplicación.	
Métodos de acción	execute	Obtiene el usuario actual y sus permisos, incluyendo los enlaces genéricos, así como un conjunto de enlaces de menú para cada uno de estos permisos, almacenándolos en el campo <code>links</code> . Asimismo obtiene el nombre del usuario actual y lo almacena en el campo <code>username</code> , para posterior consulta desde la vista.
Métodos de acceso a datos	<code>getLinks() : List<String></code>	
	Obtiene una lista de los enlaces almacenados en el campo <code>links</code> .	
	<code>getUsername() : String</code>	
	Obtiene el nombre del usuario actual almacenado en el campo <code>username</code> .	
Métodos auxiliares	Ninguno	
Acciones que la invocan	<code>menuAction</code>	
JSP resultantes	<code>menu.jsp</code>	

Tabla C-15: Detalle de la clase de acción `WarningAction`

WarningAction		
Descripción de funcionalidad	Clase encargada de gestionar las alertas automáticas de accesibilidad, permitiendo el listado y obtención.	
Métodos de acción	<code>getCheckpoints</code>	Obtiene la lista de puntos de verificación que contienen alertas automáticas para el sitio especificado por el campo <code>selectedSite</code> , almacenando el resultado en el campo <code>checkpointsList</code> . En caso de no especificar el sitio se volverá a la pantalla de selección de sitio.
	<code>getSites</code>	Obtiene la lista de los sitios web monitorizados por la aplicación y almacena el resultado en <code>siteList</code> .
	<code>getWarnings</code>	Obtiene la lista de alertas de accesibilidad para el sitio especificado por el campo <code>selectedSite</code> y el punto de verificación especificado en <code>selectedCheckpoint</code> , almacenando el resultado en el campo <code>warnList</code> . En caso de no especificar alguno de los campos se devolverá al usuario a la pantalla de selección de sitio.
	<code>showWarning</code>	Obtiene la incidencia identificada por el campo <code>selectedWarning</code> , almacenando el resultado en el campo <code>requestedWarning</code> . En caso de no especificar el campo se volverá a la pantalla de selección de sitio.
Métodos de acceso a datos	<code>getCheckpointsList() : List<String[]></code>	
	Obtiene la lista de cadenas de caracteres que identifican los puntos de verificación para el sitio especificado.	
	<code>getRequestedWarning() : AutomaticWarningBean</code>	
	Obtiene la alerta automática especificada.	
	<code>getSelectedCheckpoint() : String</code>	
	Obtiene el identificador del punto de verificación seleccionado para	

	poder hacer el seguimiento de la navegación.
	<code>getSelectedSite() : String</code>
	Obtiene el identificador del sitio seleccionado para poder hacer el seguimiento de la navegación.
	<code>getSiteList() : List<SiteBean></code>
	Obtiene el listado de sitios monitorizados por el sistema.
	<code>getWarnList() : List<AutomaticWarningBean></code>
	Obtiene el listado de las alertas automáticas de accesibilidad para el sitio y el punto de verificación especificados.
	<code>setSelectedCheckpoint(String)</code>
	Establece el punto de verificación para el cual se quiere obtener las alertas.
	<code>setSelectedSite(String)</code>
	Establece el sitio para el cual se quiere obtener los puntos de verificación.
	<code>setSelectedWarning(String)</code>
	Establece el identificador de la alerta que se quiere visualizar.
Métodos auxiliares	Ninguno
Acciones que la invocan	<code>showSites, showCheckpoints, showWarnings, showWarning</code>
JSP resultantes	<code>showSites.jsp, showCheckpoints.jsp, showWarns.jsp, showWarn.jsp</code>

Anexo D.

Detalle de los interceptores

A continuación se detallan los interceptores implementados en el sistema:

Tabla D-1: Interceptor dbInterceptor

Interceptor: loginInterceptor

Descripción: Comprueba que la conexión a la base de datos esté activa, en caso contrario devuelve la cadena "dbError" enviando al usuario la salida de dbError.jsp.

Tabla D-2: Interceptor loginInterceptor

Interceptor: loginInterceptor

Descripción: Comprueba que el usuario se haya autenticado en el sistema. El sistema almacena una relación de las sesiones (HTTPSession), identificando los usuarios asociados a estas. Cada vez que se realiza una llamada al sistema que incluya este interceptor se recupera esta entrada y en caso de que no exista un usuario asociado se rechaza la conexión, enviando al usuario a la página de login.

Tabla D-3: Interceptor adminInterceptor

Interceptor: adminInterceptor

Descripción: Este interceptor se incluye en las acciones restringidas a usuarios con permisos de administración. Comprueba que el usuario disponga de este permiso, enviándole en caso contrario a la acción e inicio, "start". Este interceptor siempre será usado después de "loginInterceptor", asegurando que el usuario esté en el sistema antes de llegar a él.

Tabla D-4: Interceptor autoWarningInterceptor

Interceptor: autoWarningInterceptor

Descripción: Este interceptor se incluye en las acciones restringidas a usuarios con permisos de visualización de alertas automáticas. Comprueba que el usuario disponga de este permiso, enviándole en caso contrario a la acción e inicio, "start". Este interceptor siempre será

usado después de “loginInterceptor”, asegurando que el usuario esté en el sistema antes de llegar a él.

Tabla D-5: Interceptor createIncidentesInterceptor

Interceptor: createIncidentesInterceptor

Descripción: Este interceptor se incluye en las acciones restringidas a usuarios con permisos de creación de incidencias. Comprueba que el usuario disponga de este permiso, enviándole en caso contrario a la acción e inicio, “start”. Este interceptor siempre será usado después de “loginInterceptor”, asegurando que el usuario esté en el sistema antes de llegar a él.

Tabla D-5: Interceptor dbInterceptor

Interceptor: dbInterceptor

Descripción: Este interceptor se incluye para realizar la comprobación del funcionamiento de la base de datos, en caso de que la conexión con esta no se pueda establecer se redirige al usuario a una salida que avisa de este error.

Tabla D-6: Interceptor knowledgeInterceptor

Interceptor: knowledgeInterceptor

Descripción: Este interceptor se incluye en las acciones restringidas a usuarios con permisos de gestión de conocimiento. Comprueba que el usuario disponga de este permiso, enviándole en caso contrario a la acción e inicio, “start”. Este interceptor siempre será usado después de “loginInterceptor”, asegurando que el usuario esté en el sistema antes de llegar a él.

Tabla D-7: Interceptor showIncidentesInterceptor

Interceptor: showIncidentesInterceptor

Descripción: Este interceptor se incluye en las acciones restringidas a usuarios con permisos de visualización de alertas automáticas. Comprueba que el usuario disponga de este permiso, enviándole en caso contrario a la acción e inicio, “start”. Este interceptor siempre será usado después de “loginInterceptor”, asegurando que el usuario esté en el sistema antes de llegar a él.

Anexo E.

Detalle de clases auxiliares

Las clases auxiliares son un conjunto de elementos que implementan funcionalidad necesaria desde múltiples partes del código. A continuación se detallan estas clases:

Tabla E-1: Detalle del enum `EnumMessages`

EnumMessages (enum)	
Descripción de funcionalidad	Enumerador que recoge un identificador para cada una de las cadenas de caracteres internacionalizadas, permitiendo a las clases referenciarlas de forma más clara y evitando errores en tiempo de ejecución. Todos los elementos de este enumerado se crean dentro de la propia clase y el constructor tiene una visibilidad <code>private</code> por lo que no se podrán crear objetos exteriores. Esta clase es utilizada por todas las clases de acción e interceptores que envían mensajes al usuario.
Métodos	<code>getMsg() : String</code>
	Obtiene la cadena de caracteres que representa al enumerado.
	<code>toString() : String</code>
	Obtiene la cadena de caracteres que representa al enumerado.

Tabla E-2: Detalle de la clase `InitSessionFactory`

InitSessionFactory	
Descripción de funcionalidad	Clase no instanciable que presenta el método estático <code>getCurrentSession()</code> , el cual se encarga de realizar una llamada al método <code>getCurrentSession()</code> de la clase <code>SessionFactory</code> de Hibernate, inicializada en la creación de <code>InitSessionFactory</code> . De esta forma todas las peticiones del sistema para obtención de la una sesión pasan a través del mismo objeto <code>SessionFactory</code> , lo que evita múltiples inicializaciones del sistema y errores de Hibernate debido a la realización de múltiples inicializaciones simultaneas.
Métodos	<code>getCurrentSession() : Session</code>
	Obtiene el objeto <code>Session</code> de <code>SessionFactory</code> .

Tabla E-3: Detalle de la clase SessionContainer

SessionContainer	
Descripción de funcionalidad	Clase que implementa un almacén de objetos <code>HTTPSession</code> e identificadores de usuario con el objetivo de conocer durante toda la sesión el usuario asociado, manteniendo sus permisos y evitando una continua reautenticación en cada acción que requiriera de privilegios. Para implementar esta clase se ha hecho uso de un patrón Singleton, el cual hace accesible a este objeto desde toda la aplicación al mismo tiempo que evita la creación de más objetos de esta clase.
Métodos	<code>getInstance() : SessionContainer</code>
	Obtiene la cadena de caracteres que representa al enumerado.
	<code>getUser(HTTPSession) : int</code>
	Obtiene el identificador de usuario asociado al objeto <code>HTTPSession</code> pasado por parámetros.
	<code>isLogged(HTTPSession) : boolean</code>
	Comprueba si el objeto <code>HTTPSession</code> está asociado a algún usuario.
	<code>registerSession(HttpSession, userId) : boolean</code>
	Almacena la sesión pasada por parámetros, asociándola con el identificador de usuario.
	<code>unRegisterSession(HttpSession) : boolean</code>
	Elimina del almacén la sesión pasada por parámetros.

Tabla E-4: Detalle de la clase Util

Util	
Descripción de funcionalidad	Clase de utilidades, creada para almacenar todos los métodos que no deban estar recogidos dentro de objetos instanciados. Para ello está clase es estática y final, además todos sus métodos internos son estáticos a su vez.
Métodos	<code>isInt(String) : boolean</code>
	Comprueba si la cadena no es vacía y si todos sus caracteres corresponden a números de 0 a 9. Se utiliza para validar los identificadores antes de pasarlos a la función <code>Integer.parseInt()</code> , lo que asegura que el formato del identificador es correcto y además evita capturar la excepción del método <code>parseInt</code> , evitando que rompa el flujo de ejecución y el uso de excepciones <code>Runtime</code> en el flujo de ejecución, tal y como recomienda el lenguaje.

Anexo F.

Ejercicio de validación

Ejercicio de validación

Este ejercicio está orientado a validar la aplicación SIGA, Sistema Integral para la Gestión de la Accesibilidad, para ello se propondrá la realización de un conjunto de tareas y posteriormente completar un cuestionario orientado a valorar su experiencia con la aplicación.

Importante: anote la hora en que comienza y termina cada tarea de forma que al finalizar pueda indicar en el cuestionario los minutos empleados en cada una de las tareas.

Datos de acceso a la aplicación:

- Páginas de acceso:
 - Acceso público: http://163.117.129.172:8080/PFC_Struts/inputNewIncidence
 - Acceso portal de gestión: http://163.117.129.172:8080/PFC_Struts/login
- Credenciales para el portal de gestión: usuario: "admin", contraseña: "difícil".

Tarea 1: Se desea tener tres roles en la aplicación, el rol de administrador (ya creado), que deberá tener únicamente permisos de administración, el rol de experto externo que tendrá permisos de creación de incidencias y el rol de gestor, que tendrá permisos de visualización de incidencias, de visualización de alertas automáticas y de gestión del conocimiento.

Tarea 2: La aplicación deberá tener 4 usuarios, uno para el experto externo (userExt), dos para los gestores de la aplicación (user01 y user02) y otro para el administrador jefe (superUser), que tendrá permisos de administración y de visualización de alertas automáticas. Las contraseñas serán de momento iguales a los nombres de usuario, pero se les pedirá a estas personas que las cambien cuanto antes. El usuario admin deberá ser eliminado de la aplicación.

Tarea 3: El rol gestor deberá tener acceso a todos los estados de las incidencias, deberá asignarse de esta manera.

- Tarea 4: Esta aplicación se utilizará para monitorizar los sitios web <http://www.uc3m.es> y <http://www.cesya.es>, por lo tanto se deberá configurar para ello.
- Tarea 5: Se deberán crear incidencias tanto desde la interfaz pública (usuario anónimo) como desde el portal de gestión (a través del experto en accesibilidad userExt), basadas en experiencias personales en los sitios web monitorizados o inventadas por el usuario.
- Tarea 6: Como gestores de la aplicación se deberán tratar las incidencias actualmente introducidas, clasificándolas, estableciendo su importancia, cambiando su estado, corrigiendo datos o insertando comentarios en función de las necesidades.
- Tarea 7: Se deberá realizar un seguimiento de los problemas de accesibilidad encontrados en los sitios web monitorizados, comprobando los puntos de verificación en los que falla cada uno y los errores específicos encontrados para estos puntos.

Anexo G.

Cuestionarios de validación

Cuestionario de validación

Por favor, rellene el siguiente cuestionario en base a sus experiencias en el manejo de la aplicación:

1. **Tiempos para la realización de las tareas** (en caso de no haber podido completar la tarea indíquelo mediante un guión en su casilla).

	Tarea 1	Tarea 2	Tarea 3	Tarea 4	Tarea 5	Tarea 6	Tarea 7
Tiempo (minutos)							

2. **Valoración general de la aplicación**

Pregunta	(Completamente en desacuerdo) -2	-1	0	1	(Completamente de acuerdo) 2
Ha sido capaz de realizar las tareas solicitadas con la aplicación.					
La interacción con la aplicación ha sido sencilla y agradable.					
Ha encontrado útil la aplicación para monitorizar la accesibilidad de sitios web.					
El sistema podría integrarse con facilidad en un sitio web ya existente.					

3. Usabilidad y accesibilidad de la aplicación

Pregunta	(Completamente en desacuerdo) -2	-1	0	1	(Completamente de acuerdo) 2
La navegación por la aplicación ha sido clara y sencilla.					
El interfaz de la aplicación tiene una estructura lógica y clara.					
A su juicio, la aplicación no presenta problemas de usabilidad.					
A su juicio, la aplicación no presenta problemas de accesibilidad.					

4. Manual de usuario

Pregunta	(Completamente en desacuerdo) -2	-1	0	1	(Completamente de acuerdo) 2
Ha sido necesario el manual de usuario para completar las tareas solicitadas.					
En caso de haber necesitado el manual de usuario, este ha sido útil para conocer el procedimiento para realizar las tareas.					
El manual de usuario tiene una estructura lógica y clara					

5. Observaciones